

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Tvorba internetové prezentace s využitím responzivního designu

Design of the Internet Presentation Using a Responsive Design

Student: Jakub Kohoutek

Vedoucí bakalářské práce: Mgr. Jaroslav Ševčík, Ph.D.

Ostrava 2015

Zadání bakalářské práce

Student:

Jakub Kohoutek

Studijní program:

B6209 Systémové inženýrství a informatika

Studijní obor:

6209R025 Systémové inženýrství a informatika

Téma:

Tvorba internetové prezentace s využitím responzivního designu
Design of the Internet Presentation Using a Responsive Design

Zásady pro vypracování:

1. Úvod
2. Teorie a metody tvorby webových stránek
3. Koncepce webové prezentace
4. Responzivní design a jeho implementace
5. Nasazení webové prezentace
6. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků bakalářské práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

HAUSER, M., T. HAUSER a Ch. WENZ. *HTML a CSS - Velká kniha řešení*. Brno: Computer Press, 2006. 912 s. ISBN 80-251-1117-2.

HOGAN, Brian. *HTML5 a CSS3 – výukový kurz webového vývoje*. Brno: Computer Press, 2011. 272 s. ISBN 978-8025-135-761.

KADLEC, Tim. *Responzivní design – profesionálně*. Brno: Zoner Press, 2014. 248 s. ISBN 978-80-7413-280-3.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Jaroslav Ševčík**

Datum zadání: 21.11.2014

Datum odevzdání: 07.05.2015

doc. Ing. Jana Hančlová, CSc.
vedoucí katedry



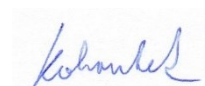
prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Místopřísežné prohlášení o samostatném vypracování bakalářské práce

„ Prohlašuji, že jsem celou práci, včetně všech příloh vypracoval samostatně“.

Zároveň bych na tomto místě chtěl poděkovat svému vedoucímu bakalářské práce, Mgr. Jaroslavu Ševčíkovi, Ph.D. především za velmi vstřícný přístup, cenné rady a připomínky, které přispěly k dokončení bakalářské práce.

Datum odevzdání bakalářské práce: 10. 7. 2015



Jakub Kohoutek

Obsah

1 Úvod.....	6
2 Teorie a metody tvorby webových stránek	7
2.1 Internet	7
2.1.1 Historie Internetu.....	7
2.1.2 World Wide Web	7
2.1.3 W3C	8
2.2 HTML	8
2.2.1 Historie HTML.....	8
2.2.2 XHTML.....	9
2.2.3 HTML5	10
2.2.4 Srovnání HTML 4.01 s HTML5	11
2.2.5 Problém s HTML5	13
2.3 Kaskádové styly CSS	14
2.3.1 Historie CSS.....	14
2.3.2 Připojení stylů ke stránce	15
2.3.3 Struktura stylového předpisu.....	17
2.3.4 Pseudoprvky a pseudotřídy	18
2.3.5 Dědičnost.....	20
2.3.6 Kaskáda	20
2.3.7 CSS3.....	21
2.4 Dynamické webové stránky	22
2.4.1 PHP	22
2.4.2 MySQL.....	23
2.4.3 JavaScript	24
2.5 Webové prohlížeče.....	24

2.6 HTML editory	27
2.6.1 Strukturní editory	27
2.6.2 WYSIWYG editory	27
2.6.3 Redakční systémy.....	27
3. Koncepce webové prezentace	28
3.1 Produkční plán.....	28
3.1.1 Název webu	28
3.1.2 Záměr a cíle.....	28
3.1.3 Cílové publikum.....	28
3.1.4 Vzory návštěvnosti.....	29
3.1.5. Příjmový model.....	29
3.1.6 Sada schopností	29
3.1.7 Konkurence	29
3.1.8 Marketingový plán	30
3.1.9 Tým	30
3.1.10 Předpoklady.....	30
3.2 Základní části webu.....	30
3.2.1 Záhloví.....	30
3.2.2 Horizontální navigační menu	31
3.2.3 Levý sloupec	31
3.2.4 Hlavní část.....	32
3.2.5 Zápatí.....	32
3.3 Grafická podoba webu	32
3.4 HTML kód	33
3.5 CSS kód.....	35
3.6 Kniha návštěv	37
4. Responzivní design a jeho implementace	38
4.1 Proč responzivní design	38

4.2 Plánování	38
4.2.1 Faktory ke zvážení	38
4.2.2 Analýzy	39
4.2.3 Testovací prostředí	40
4.3 Jednotky	41
4.4 Mediální dotazy	41
4.4.1 Viewporty	42
4.4.2 Pořadí mediálních dotazů	42
4.4.3 Breakpointy	43
4.4.4 Navigace	45
4.5 Responzivní obsah	46
4.6 Responzivní obrázky	47
5. Nasazení webové prezentace	47
5.1 Validátory	47
5.2 SEO	48
5.3 Jak dostat web do vyhledávačů	49
5.4 Webhosting	50
6. Závěr	52
Seznam použité literatury	53
Seznam zkratk	54
Prohlášení o využití výsledků bakalářské práce	
Seznam příloh	
Přílohy	

1 Úvod

V dnešní době má k internetu přístup snad většina obyvatel civilizovaného světa. Dnes již není nic neobvyklého dostat se k internetu i v zemích Afriky, kde to bylo ještě před pár lety nepředstavitelné. Je jen otázkou pár let, než se připojení dostane na zbytek míst, kde jej zatím nenajdeme. To vše svědčí o velkém potenciálu a zájmu o internet.

Internet se stal součástí našeho života a to díky jeho širokému spektru využití. Je součástí mnoha oborů, stal se důležitým pomocníkem při práci, zábavě či komunikaci. Obrovskou výhodou internetu je rychlost přenosu dat a překonání vzdálenosti. Díky internetu se dnes lidé mohou spojit a komunikovat ať už se nachází kdekoli na světě. Výhod internetu se samozřejmě snaží využít firmy od velkých společností až po malé živnostníky k prezentaci své činnosti. Některé se snaží pouze o svou prezentaci, jiné zas o získání nových zákazníků, prodej produktů či služeb. Využití internetu je opravdu široké, a proto je dnes velkou nevýhodou absence internetové prezentace.

S vývojem technologií roste také počet zařízení, na kterých se dá internet využívat. V současné době už asi každý vlastní chytrý mobilní telefon, umožňující vyřizovat emaily či prohlížet internetové stránky. Existuje mnohem více zařízení, na kterých se dá internet využít, například tablety, nebo chytré televizory. Problém je, že tyto zařízení mají různé rozlišení, a proto se daná internetová stránka nemusí zobrazit zrovna korektně. To poté může odradit zákazníky webu a už jej nemusí znova navštívit. K tomu aby se internetová stránka zobrazila správně, na co nejvíce zařízeních slouží responzivní design.

Cílem této práce je vytvořit internetovou prezentaci s využitím responzivního designu. Internetová prezentace bude tvořena pro kadeřnictví, které ještě nikdy svou vlastní prezentaci na internetu nemělo. Kadeřnictví má v blízkosti konkurenci, která svou internetovou prezentaci má, ale ne v moc dobrém stavu. Proto bude snaha této práce o vytvoření přehledné, kvalitní a validní internetové prezentace. Důležitá bude také SEO optimalizace, aby se pozice kadeřnictví po zavedení postupně dostávala co nejvýše ve výsledcích nejznámějších vyhledávačů.

Tato práce bude zaměřená na využití responzivního designu. Bude popsáno proč se na responzivní design zaměřit, dále princip tvorby responzivního designu a jeho funkce. Samotná internetová prezentace bude tvořena tak, aby se správně zobrazovala na dnes nejpoužívanějších zařízeních.

2 Teorie a metody tvorby webových stránek

2.1 Internet

Internet je označením pro celosvětový systém navzájem propojených počítačových sítí. Umožňuje komunikaci mezi zařízeními připojených k této síti, bez ohledu na jejich vzájemnou vzdálenost. Komunikace probíhá pomocí sady pravidel, takzvaných protokolů rodiny TCP/IP.

2.1.1 Historie Internetu

Internet vznikl díky americké armádě, která potřebovala, aby při případném vojenském útoku a možném výpadku uzlu zůstala síť i nadále funkční. Proto vytvořila síť ARPANET (Advanced Research Project Agency Net), která byla vybudována jako síť s navzájem nezávislými uzly. Síť byla zprovozněna roku 1969 a tento rok se uvádí také jako rok vzniku Internetu, který však měl k dnešnímu Internetu daleko. Počátkem 70. let se síť ARPANET rychle rozrůstala a připojily se do ní i univerzity v USA. V roce 1973 se stala síť mezinárodní, z důvodů připojení univerzit a institucí mimo USA. Poté byli vyvinuty další sítě na stejném principu jako pracoval ARPANET. Tyto sítě spolu ale navzájem nekomunikovaly. Proto Bob Kahn a Vint Cerf vymysleli novou technologii TCP/IP. Díky protokolům TCP/IP bylo umožněno propojit vzdálené a různorodé sítě a tak vznikla velká síť sítí, nazývaná Internet. (M., T. Hauser, Wenz, 2006).

2.1.2 World Wide Web

Pod slovem Internet si většina lidí představuje právě to, co je ve skutečnosti World Wide Web (WWW) neboli celosvětově propojená pavučina obsahující milióny stránek s nejrůznějším obsahem. Na většinu těchto stránek se lze připojit poklepáním na jejich odkaz, či napsáním jejich adresy, ať už se nachází na kterémkoli serveru na světě (Písek, 2014).

Přestože je World Wide Web v současné době nejrozšířenější službou Internetu a hodně lidí si tuto službu ztotožňuje se samotným Internetem, je jeho relativně novou částí. Službu vyvinuli na konci 80. let výzkumníci z Evropské laboratoře pro fyziku částic (CERN) ve Švýcarsku, kteří si chtěli zjednodušit přístup ke svým výzkumným dokumentům. Mezi výzkumníky byl i Tim Berners-Lee, který je považován za vynálezce webu. Tim Berners-Lee se následovně postaral o představení textového prohlížeče a vyvinutí značkovacího jazyka HTML. Po představení služby World Wide Web uživatelům Internetu se rozpoutala revoluce v jeho vývoji (Písek, 2014).

2.1.3 W3C

Mezinárodní konsorcium W3C (World Wide Web Consortium) vzniklo roku 1994 a jeho zakladatelem je Tim Berners-Lee, který je stále jeho předsedou. Členové konsorcia se společně s veřejností starají o vývoj standardů pro World Wide Web. V době rozmachu webu mohl mít téměř každý webový vývojář svou verzi HTML, a proto byly vymyšleny a neustále se zdokonalují tyto standardy.

Web w3.org uvádí, že cílem konsorcia je „Rozvíjet World Wide Web do jeho plného potenciálu vývojem protokolů a směrnic, které zajistí dlouhodobý růst Webu“. Mezi jednotlivé aspekty cíle patří hlavně web pro všechny a web na všem (World wide web consortium, 2015).

2.2 HTML

HTML (HyperText Markup Language) je značkovací jazyk, který se používá pro tvorbu internetových stránek. Jednotlivé stránky jsou propojeny hypertextovými odkazy. Značkovací jazyk obohacuje prostý text o další informace prostřednictvím značek. Veškeré značky neboli tagy HTML jazyka se uzavírají do špičatých závorek a jsou určeny pouze pro prohlížeč. To, co není uzavřeno v těchto závorkách, je text, který vidí uživatel ve svém prohlížeči.

HTML poskytuje párové a nepárové značky:

- párové značky tvoří pár - první značka něco začíná a druhá, uzavírací, končí, uzavírací značka obsahuje vždy lomítko, před názvem značky,
- nepárová značka definuje nějaký prvek, má vliv sama na sebe.

2.2.1 Historie HTML

HTML prošlo několika fázemi, přičemž první verze HTML, 1.0, se objevila v roce 1990. V roce 1993 se představila neoficiální verze HTML+, která ale neobsahovala formátování odstavců ani úpravu textu. Mnohé z elementů, které byly součástí HTML+ již dnes v HTML nenajdeme. První verze, která měla formální specifikaci a stala se oficiálním standardem, byla verze HTML, 2.0, uvedená v roce 1994. Tato verze měla dvě úrovně, ta první pouze rozšiřuje první verzi HTML, druhá úroveň obsahovala práci s formuláři. Již na jaře 1995 byla vydaná verze HTML, 3.0. Tuto verzi ale nepodporoval téměř žádný prohlížeč a tak byla v roce 1996 uvedená na trh verze HTML, 3.2, která byla považována za pravého nástupce HTML, 2.0. Na jaře roku 1997 se začalo pracovat na verzi HTML, 4.0, které se během jejího vývoje přezdívalo Cougar. V prosinci 1997 bylo HTML, 4.0, přijato jako standard organizací W3C. Tato verze již podporovala kaskádové styly, skriptování na straně klienta, nebo například rozpoznání jazyků. Od té doby se vývoj jazyka HTML zpomalil, takřka zastavil. V roce 1999

byla vydána verze HTML, 4.01, která jen opravuje menší chyby předchozí verze. O nejnovější verzi HTML5 věnovaná kapitola 2.2.3 (Písek, 2014).

2.2.2 XHTML

Programátoři používají pro tvorbu svých vlastních značek jazyk XML (Extensible Markup Language) v překladu rozšiřitelný značkovací jazyk. Organizace W3C přetvořila jazyk HTML pomocí použití XML na místo původního SGML jazyka používaného pro HTML a nazvala ho XHTML (Extensibel HTML). V roce 2000 byla tedy vydána první verze XHTML, 1.0 která měla tři verze:

- XHTML 1.0 Strict,
- XHTML 1.0 Transitional,
- XHTML 1.0 Frameset.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <title>XHTML 1.0 Strict dokument</title>

  </head>
  <body>

  </body>
</html>
```

Obr. 2.1 XHTML dokument (Zdroj: autor)

O rok později byla vydána, jako doporučení organizace W3C verze XHTML, 1.1. I když je XHTML takzvaným rozšířeným jazykem, je ve své funkčnosti prakticky stejným jako HTML. Na rozdíl od HTML ale netoleruje chyby, například pokud je v HTML zapomenuto na uzavírací tag, nic se nestane. V XHTML by to nefungovalo (Wempen, 2007).

Další rozdíly jsou například:

- nutnost zápisu tagů a jejich atributů malými písmenky,
- nutnost zadávat hodnoty atributů do uvozovek,
- nutnost ukončovat i nepárové tagy a to lomítkem,
- nutnost, aby všechny atributy měli zadanou hodnotu,
- nutnost, aby byli párové tagy párovými povinně (Dušan Janovský, 2014).

Původně se počítalo s tím, že se XHTML stane nástupcem HTML. Organizace W3C pracovala ještě na nové verzi XHTML, 2.0. Ta ale nebyla kompatibilní s předchozími verzemi HTML a XHTML a to bylo pro výrobce prohlížečů nepřijatelné. To nakonec uznala i W3C, a proto začalo v roce 2007 vznikat HTML5 (Jiří Kosek, 2013).

2.2.3 HTML5

Gasston (2015, str. 26) uvádí, že: „*HTML5 je ve skutečnosti snaha o vývoj webu směrem, jenž by pokryl většinu současných požadavků, které se neuvěřitelně změnily od počáteční představy jednoduché sítě vzájemně propojených dokumentů.*“

HTML5 je následníkem specifikace HTML 4.01 a konsorcium W3C jej předvedlo do stavu kandidátního doporučení v červenci roku 2014. HTML5 obsahuje nové funkce, další prohlašuje za zastaralé nebo už v této specifikaci vůbec nejsou. Dále jsou přidány nové elementy a je zjednodušen zápis některých stávajících. HTML5 umožňuje stále používat pro zápis syntaxi XML (Gasston, 2015).

Další hlavní změny:

- DOCTYPE – zkrácený zápis specifikace typu dokumentu,
- zjednodušený zápis jazyku a kódování dokumentu,
- zvýšení přehlednosti zdrojového kódu,
- několik nových funkcí,
- multimediální obsah,
- nové atributy formulářů.

2.2.4 Srovnání HTML 4.01 s HTML5

V HTML5 došlo k mnoha zjednodušením, na některé z nich můžeme narazit hned v hlavičce dokumentu. Hned první řádek s typem dokumentu je v HTML5 mnohem jednodušší (Písek, 2014).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Obr. 2.2 - příklad dřívějšího zápisu DOCTYPE

```
<!DOCTYPE html>
```

Obr. 2.3 - zápis DOCTYPE v HTML5

Také údaje o použité znakové sadě se nyní zapisují mnohem snadněji a přehledněji.

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Obr. 2.4 - příklad dřívějšího zápisu údajů o znakové sadě

```
<meta charset="utf-8">
```

Obr. 2.5 – zápis údajů o znakové sadě v HTML5

Dále lze uvést do tagu <html> v jakém jazyce je dokument napsán nebo také jaké zarovnání písma je použito. U češtiny, kde píšeme zleva doprava, to bude vypadat následovně <html lang="cs" dir="ltr">, v případě opačného zarovnání písma by namísto "ltr" (left to right) byla uvedena zkratka "rtl", což znamená right to left (Písek, 2014).

```
<!DOCTYPE html>
<html lang="cs" dir="ltr">
  <head>
    <title>Dokument HTML5</title>
    <meta charset="utf-8">
    <meta name="description" content="">
    <meta name="keywords" content="">
    <meta name="author" content="">
    <meta name="robots" content="all">
  </head>
  <body>
    <h1>Toto je HTML5 dokument</h1>
  </body>
</html>
```

Obr. 2.6 – HTML5 dokument

Další a velmi významnou změnou jazyka HTML5 jsou nové elementy, jejichž cílem je dát stránkám lepší, přehlednější strukturu. Vývojářům tato novinka přinese více možností v označování oblastí a obsahu webové stránky. Dříve byly stránky takzvaně "předivovány",

obsahovaly spoustu elementů div s atributy id nebo class. Element div je blokový prvek, který nenese žádný význam. Slouží k obalení části webu, na kterou je potřeba uplatnit určitý stylový předpis. Pomocí divů byly tvořeny veškeré části webu, jako je záhlaví, navigační menu, různé postranní sloupce, prostor pro hlavní obsah či články a zápatí webu (Gasston, 2015).

```
<div id="footer">Toto je obsah zápatí dokumentu</div>
```

Obr. 2.7 – tvorba zápatí v HTML 4.01

```
<footer>Toto je obsah zápatí dokumentu</footer>
```

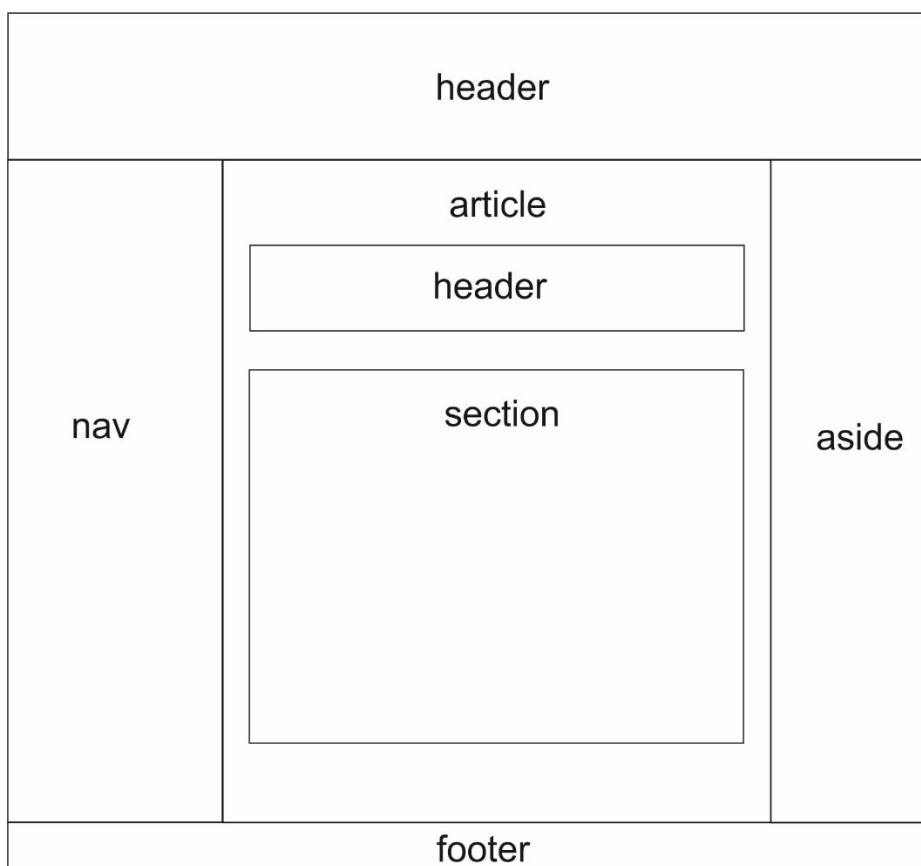
Obr. 2.8 – tvorba zápatí v HTML5

HTML5 obsahuje deset strukturálních elementů, první tři již známe, další čtyři označujeme jako rozděľující elementy a poslední tři definují oblasti uvnitř rozděleného obsahu:

- body – tělo dokumentu,
- h1-h6 – nadpisy šesti úrovní, podle váhy a velikosti,
- address – kontaktní informace k dokumentu, vhodný pro využití v patičce dokumentu, k zadání jmen autorů dokumentu, nebo společnosti, která jej vytvořila,
- article – nezávislá část dokumentu, například článek, příspěvek blogu, diskuzního fóra nebo komentář, tento element může obsahovat části header, section, footer,
- aside – okrajově souvisí s okolním obsahem, lze použít i za nezávislou část, například postranní sloupek,
- nav – navigační oblast, obsahuje odkazy na další dokumenty či oblasti stejného dokumentu,
- section – tematické seskupení obsahu, například úvod na domovské stránce, nebo kapitola knihy,
- footer – zápatí dokumentu,
- header – záhlaví dokumentu,
- hgroup – seskupuje více nadpisů.

Tyto nové elementy vznikly za účelem rozčlenit dokument tak, aby ho mohli snadněji parsovat webové prohlížeče nebo například předčítače obrazovek pro zrakově postižené uživatele (Gasston, 2015).

Na obrázku 2.9 můžeme vidět příklad rozložení stránky pomocí strukturálních elementů v rámci standardu HTML5. Nové elementy pěkně vyniknou na příkladu kdysi typického třísloupcového layoutu. Ten tvoří záhlaví, levý navigační sloupec s odkazy, uprostřed hlavní část stránky a vpravo doplňující sloupec, celé rozložení je zakončeno zápatím.



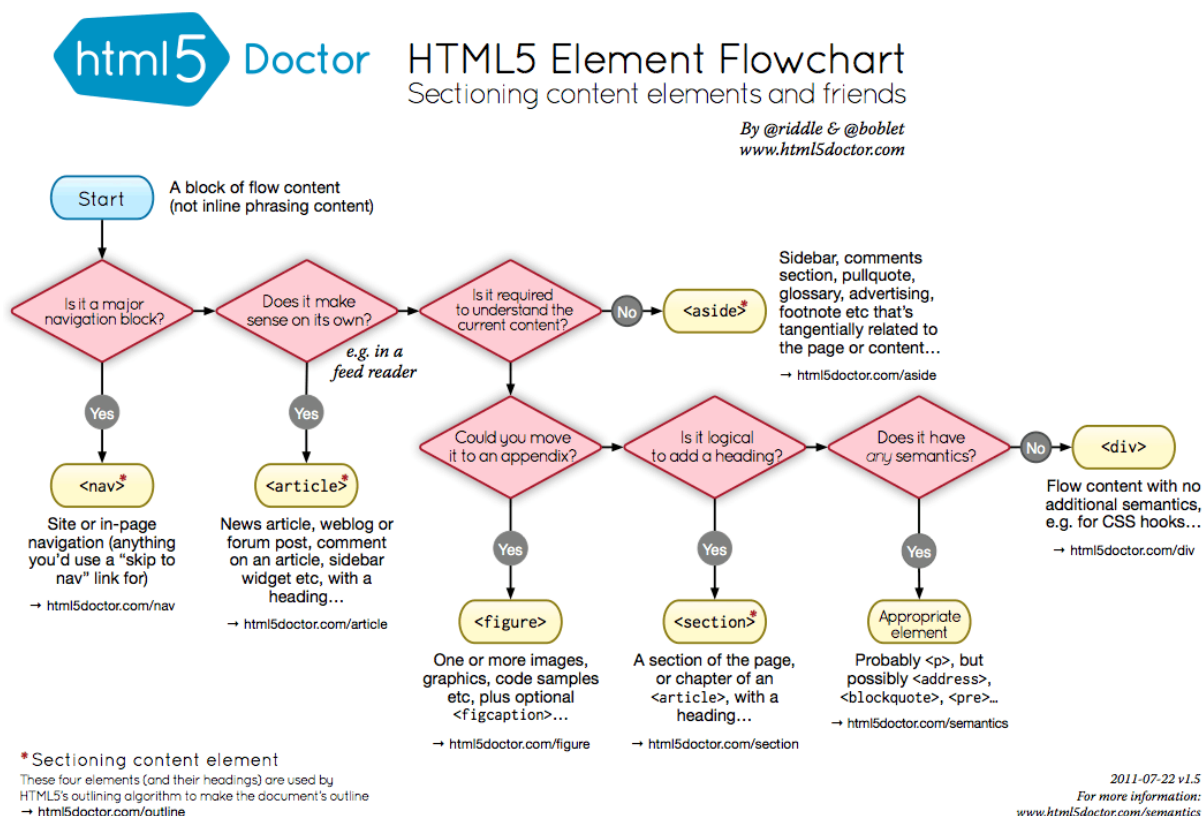
Obr. 2.9 – příklad rozložení dokumentu HTML5 (Zdroj: autor)

2.2.5 Problém s HTML5

Zacházení s některými elementy může být problematické. Například rozdíl mezi elementy section a article je těžko rozpoznatelný. Sekce může obsahovat element article a další sekce, article může obsahovat element section a další elementy article. Rozdíly mezi nimi jsou, ale ani samotný autor specifikace je ještě nedokázal nějakým jednoduchým způsobem specifikovat (Gasston, 2015).

Protože se objevilo mnoho lidí, kteří měli problémy rozlišit, co jednotlivé rozdělující elementy dělají, objevil se diagram, který pomáhá vybrat správný element pro danou situaci. Tento diagram vymysleli přispěvatelé na stránkách html5doctor.com.

Diagram je složen ze šesti podmínek, podle kterých vývojáři postupují a zjišťují, jaký element ze standardu HTML5 pro daný obsah použít. Jestliže podmínka vyhovuje, použije se doporučený element. Pokud žádná z podmínek nevyhovuje danému obsahu, je použit prvek `div` (Gasston, 2015).



Obr. 2.10 – Rozhodovací diagram pro výběr správného elementu (Zdroj:

<http://html5doctor.com/downloads/h5d-sectioning-flowchart.png>)

2.3 Kaskádové styly CSS

Kaskádové styly mají zkratku CSS z anglického Cascading Style Sheet. Umožňují tvořit grafickou podobu internetových stránek. Kaskádové se jmenují, protože na sebe mohou vrstvit definice stylu, ale platí jen ta poslední. Díky CSS je umožněno oddělit vzhled stránky od jejího obsahu a struktury.

2.3.1 Historie CSS

Původně měl oddělení vzhledu od struktury dokumentu umožňovat už jazyk HTML od samotných počátku v roce 1990. Berners-Lee svůj stylový předpis nikdy nepublikoval, spoléhal na autory prohlížečů, že koncepci rozvinou. To se však nestalo. Popularita webu stoupala a jejich vývojáři si stále více stěžovali na to, že nemohou vzhled HTML dokumentů ovlivnit. V roce 1994 publikoval Hakon Wium Lie první koncept kaskádových stylů. Návrh

nebyl dokonalý, ale stal se základem, ze kterého vycházejí kaskádové styly dodnes. O tři dny později po publikaci první konceptu kaskádových stylů se objevila první verze prohlížeče Netspace. Ten řešil situaci po svém a zabudoval základní formátovací prostředky přímo do jazyka HTML. To se ale ukázalo jako velmi špatné řešení, které poznamenalo kompatibilitu a použitelnost webových dokumentů na mnoho let dopředu (Prokop, 2005).

Až v prosinci roku 1996 byla jako doporučení konsorcia W3C vydána první verze kaskádových stylů CSS1. Konsorcium W3C kaskádové styly dále rozvíjelo a v květnu 1998 bylo vydáno doporučení specifikace CSS2. Následně byla vydána revize CSS 2.1, která hlavně opravovala některé chyby, ta byla dokončena v červnu roku 2011. Nejnovější verze CSS3 se začala vyvíjet už roku 2005 a jeho dokončení je plánováno na rok 2015 (Prokop, 2005).

To že se kaskádové styly stále rozvíjí, ale neznamená, že bude specifikace CSS4 – ta nebude. CSS3 je “vše po specifikaci CSS 2.1” a je plánováno, že se v budoucnu všechny rozdíly ztratí a zůstane jen jazyk CSS (Gasston, 2015).

2.3.2 Připojení stylů ke stránce

1. Atribut style – v tomto případě se styly přiřazují přímo jednotlivým elementům daného dokumentu. Tento způsob se používal dříve, dnes už se atribut style nepoužívá.

```
<body>
  <h1 style="font-size: 200%; color: #339900">Nápis dokumentu</h1>
  <p style="font-style: italic">Odstavec který bude kurzivou</p>
  <ul style="list-style-type: none">
    <li>První</li>
    <li>Druhý</li>
    <li>Třetí</li>
  </ul>
</body>
```

Obr. 2.10 – použití atributu style (Zdroj: autor)

2. Prvek <style> - stylový předpis přímo v HTML dokumentu. Musí být umístěn uvnitř hlavičky dokumentu v sekci head. Tento zápis stylů je také starší a dnes se nepoužívá. Prvek má tři atributy:

- type – určuje typ obsahu, v tomto případě má hodnotu vždy text/css,
- media – nepovinný atribut, který určuje médium pro které je určen daný stylový předpis,
- title – slouží k přiřazení titulku danému stylovému předpisu (Prokop, 2005).

```

<style type="text/css">
  body {
    background-color: #FFFF99;
    font-size: 100%;
    font-family: Arial, Helvetica;
  }
  h1 {
    font-size: 150%;
    margin-left: 10px;
  }
</style>

```

Obr. 2.11 – tabulka stylů v HTML dokumentu (Zdroj: autor)

Předešlé dva způsoby připojení stylového předpisu se již nepoužívají a to především proto, že stylový předpis zvětšuje celkový objem dokumentu a načítají se s každou novou stránkou znovu, protože se necachují. Další problém je, že jedna změna stylu se musí provést obvykle do více dokumentů, z toho vyplývá, že se dokumenty obtížně udržují (Prokop, 2005).

3. Prvek `<link>` - slouží k připojení tabulky stylů uložené v externím souboru. Jedná se o standardní způsob připojování stylového předpisu. Prvek musí být umístěn uvnitř hlavičky dokumentu v sekci head. Tento prvek má pět atributů:

- href – cesta k externímu souboru se styly,
- type - určuje typ obsahu externího souboru, v tomto případě má hodnotu vždy text/css,
- rel – určuje, jaká je vazba dokumentu k externímu zdroji, stylesheet v případě základních stylových předpisů a alternate stylesheet v případě alternativních stylových předpisů,
- media - nepovinný atribut, který určuje médium pro které je určen daný stylový předpis,
- title - slouží k přiřazení titulku danému stylovému předpisu (Prokop, 2005).

Tento způsob připojování stylového předpisu má řadu výhod například:

- zjednodušená správa webu, jelikož pro celý web nebo jeho určitou část stačí pouze jeden soubor se stylovým předpisem, to umožňuje jednotnou grafickou úpravu,
- stylový předpis se cachuje na straně klienta po prvním načtení, takže se načítání dalších souvisejících dokumentů podstatně urychlí,
- oddělení formátování od struktury dokumentu, to má pozitivní význam pro přenositelnost a správu webovského obsahu (Prokop, 2005).

```

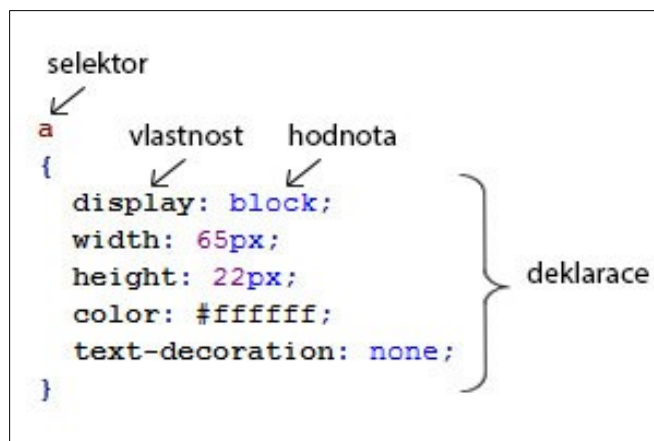
<head>
  <title>Externí stylopis</title>
  <meta charset="utf-8">
  <meta name="description" content="externí stylopis">
  <link rel="stylesheet" type="text/css" href="style.css" >
</head>

```

Obr. 2.12 – záhlaví dokumentu s připojením externího stylového předpisu (Zdroj: autor)

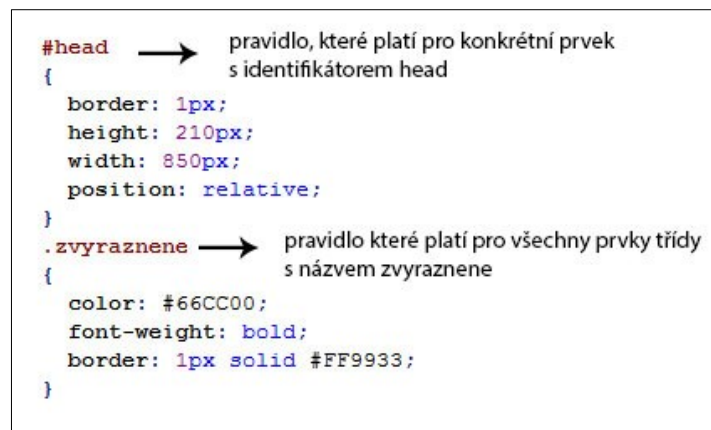
2.3.3 Struktura stylového předpisu

Stylový předpis je definován jedním či více pravidly, která určují, jak by měl být vybraný prvek zobrazen. Jako první se zapisuje selektor, který určuje, na které prvky se má styl použít. Jako selektor lze použít prvek HTML jazyka, nebo hvězdička, která znázorňuje, že styl platí pro všechny prvky. Následuje deklarace, která se skládá z vlastnosti a její hodnoty.



Obr. 2.13 – struktura stylového předpisu (Zdroj: autor)

Velmi často se používají identifikátory id a třídy class. Identifikátor id jednoznačně pojmenovává nějaký prvek v HTML dokumentu. Prvek s tímto id se může v dokumentu vyskytovat pouze jednou. Poté je možno přidat pravidlo, které bude platné pouze pro jeden konkrétní prvek dokumentu. Ve stylovém předpisu je identifikátor označen hashtagem a názvem identifikátoru. Atribut class zařazuje prvek nebo více prvků do třídy. Každý prvek může být ve více třídách a každá třída může obsahovat více prvků. Ve stylovém předpisu je třída označena tečkou a jménem třídy.



Obr. 2.14 – označení id a class ve stylovém předpise (Zdroj: autor)

Kaskádové styly dovolí upřesnit kontext, v němž se musí popisovaný prvek vyskytovat, aby pravidlu vyhověl. Mezi nejpoužívanější patří:

- X Y znamená prvek Y, který je obsažen v prvku X
- X>Y znamená prvek Y, který je přímým potomkem X
- X+Y znamená prvek Y bezprostředně předcházený sourozencem X

2.3.4 Pseudoprvky a pseudotřídy

Pseudoprvky či pseudoelementy se používají k přidání efektů k určitým HTML elementům. Prohlížeč je vloží do určitého elementu na základě nějakého stavu. Ve stylovém předpisu je pseudoprvek označen dvojtečkou, nebo dvěma dvojtečkami, které by v budoucnu měly být stoprocentně podporovány ve všech internetových prohlížečích (Josef Žáček, 2014).

Typy CSS pseudoprvků:

- :before – vkládá generovaný obsah před začátek daného prvku
- :after – vkládá generovaný obsah za konec obsahu daného prvku
- :first-letter – aplikuje daný styl na první písmeno obsahu prvku, hodí se k různým typografickým úpravám, například na iniciály,
- :first-line – aplikuje daný styl na první formátovaný řádek odstavce, lze aplikovat pouze na blokové elementy (Prokop, 2005).

```

p:before → vygeneruje text Novinka! na začátku každého odstavce
{
  content: "Novinka!";
}
p:first-letter → nastaví první písmeno každého odstavce větší a tučným písmem
{
  font-size: 200%;
  font-weight: bold;
}
p:first-line → první řádky všech odstavců budou zobrazeny verzálkami
{
  text-transform: uppercase;
}

```

Obr. 2.15 – příklady pseudoprvků (Zdroj: autor)

Pseudotřídy jsou aktivovány obvykle nějakou událostí nebo stavem, rozlišují prvky podle jiných charakteristik než je jméno, atributy, či obsah. Prvek může střídavě patřit a nepatřit k dané pseudotřídě, podle toho, co uživatel s dokumentem provádí. Pseudotřídy jsou aplikovány vždy na celý obsah prvku, na rozdíl od pseudoprvků které jsou aplikovány jen na část obsahu daného prvku (Prokop, 2005).

Pseudotřídy se používají nejvíce pro odkazy. Některé jsou statické a jejich použití je rozhodnuto už při načtení stránky, další zase dynamické a jejich použití je vyvoláno určitou událostí. Typy pseudotříd:

- `:link` – vztahuje se k nenavštíveným odkazům, pro odlišení vzhledu od navštíveného odkazu,
- `:visited` – vztahuje se k již navštíveným odkazům,
- `:hover` – tato pseudotřída se aktivuje a působí na prvek v okamžiku, kdy uživatel najede kursor myši na daný prvek, ve většině případů odkaz,
- `:active` – působí na prvek v okamžiku, kdy je prvek aktivován, to znamená například v okamžiku mezi zmáčknutím a uvolněním tlačítka myši,
- `:focus` – působí na prvek v okamžiku, kdy byl zaměřen, například pomocí klávesy tabulátor, nebo vstupní pole formuláře s textovým kurem,
- `:first-child` – působí na všechny prvky, které jsou prvním potomkem jiného prvku,
- `:lang` – působí na všechny prvky daného jazyka, například `:lang(en)` vybere všechny prvky psané anglicky (Prokop, 2005).

<code>a:hover</code>	→	odkaz se zvětší a zbarví do černa v okamžiku kdy na něj najede kurzor
<pre>{ font-size: 150%; color: black; }</pre>		
<code>a:visited</code>	→	navštívený odkaz bude fialový
<pre>{ color: purple; }</pre>		
<code>p:first-child</code>	→	všechny odstavce, které jsou "prvním dítětem" jiného prvku budou tučným písmem
<pre>{ font-weight: bold; }</pre>		

Obr. 2.16 – příklady pseudotříd (Zdroj: autor)

2.3.5 Dědičnost

Dříve než se u prvku použije výchozí hodnota, pokusí se prvek vlastnost zdědit, to znamená, že ji převezme od svého rodičovského prvku. Nemá-li prvek vlastnost definovanou, zdědí se po rodiči, jestli-že nemá ani rodič vlastnost definovanou, pak se zdědí po prarodiči. Teprve když nemá vlastnost definovanou ani prarodič, použije se u prvku výchozí hodnota. Dědit lze pouze vlastnosti popisující styl obsahu, tedy barva a typ písma, formát textu a podobně.

2.3.6 Kaskáda

Kaskádové styly umožňují definovat vlastnosti každého prvku několikrát, lze připojit více tabulek stylů a prvek může vyhovovat více pravidlům. Proto byly vytvořeny přesná pravidla, která popisují, jak více definic stylů pro jeden prvek zpracovat, která pravidla mají přednost a jak se mohou doplňovat.

Porovnává se konkrétnost selektoru v každém pravidle, čím konkrétnější je selektor tím víc má bodů a větší přednost:

- názvy prvků – nejobecnější popis – 1 bod,
- třídy, pseudotřídy a jiné atributy – konkrétnější popis – 10 bodů,
- identifikátory prvků – nejvíce konkrétní – 100 bodů,
- styly definovány přímo v prvku – absolutní přednost – 1000 bodů.

Pro výpočet konkrétnosti selektoru jsou stanoveny tři hodnoty:

- a = počet ID v selektoru,
- b = počet ostatních atributů v selektoru
- c = počet názvů prvků v selektoru.

Tři získaná čísla se zapíší za sebou a získané číslo udává konkrétnost selektoru.

Příklady:

* {}	=	a=0 b=0 c=0	→	konkrétnost = 0
p {}	=	a=0 b=0 c=1	→	konkrétnost = 1
ul li {}	=	a=0 b=0 c=2	→	konkrétnost = 2
ul li.top {}	=	a=0 b=1 c=2	→	konkrétnost = 12
li.top.red {}	=	a=0 b=2 c=1	→	konkrétnost = 21
#main {}	=	a=1 b=0 c=0	→	konkrétnost = 100

Největší přednost mají pak pravidla označené klíčovým slovem !important. Toto klíčové slovo se píše za deklaraci pravidla. Pokud je tímto klíčovým slovem označená některá sdružená vlastnost jako například border, má to stejný účinek, jako by byly označeny i všechny její dílčí vlastnosti, jako například border-color, border-radius (Prokop, 2005).

2.3.7 CSS3

CSS3 zavádí spoustu nových funkcí a standardizuje již existující funkce implementovány odlišně v různých webových prohlížečích, aby byl jazyk CSS schopen přežít ve světě, kde je možno zabudovat webový prohlížeč takřka kdekoli (Gasston, 2015).

Specifikace CSS3 je modulární, což znamená, že se skládá z několika menších specifikací, které mohou webové prohlížeče implementovat samostatně. Není proto specifikací na jeden dokument jako tomu bylo u specifikace CSS 2.1 (Gasston, 2015).

Revoluci způsobily dotazy na médium (media queris), což jsou speciální konstrukce, které umožňují styly aplikovat podle rozměrů obrazovky a schopností webového prohlížeče. Pomocí media queris lze zjistit, jaké schopnosti má zobrazovací zařízení kterému je stránka odesílána a podle toho je mu odeslán styl, který je pro něj vhodný. To byl významný krok umožňující stylování pro více zařízení (Gasston, 2015).

2.4 Dynamické webové stránky

Dynamické webové stránky mění svůj vzhled, či obsah v závislosti na čase, aktivitě návštěvníka a podobně. Dynamické stránky umožňují například odesílat data prostřednictvím formulářů, vyhledávání na webu, generování obsahu na žádost uživatele nebo reagují na pohyb kursoru myši. Aby byl web dynamický, využívají se skriptovací jazyky. Ty pracují na straně serveru, jako například PHP, nebo na straně klienta a jeho internetového prohlížeče, jako například JavaScript. Opakem dynamických stránek jsou statické webové stránky, u kterých se obsah nemění. Uživatelé získají jejich obsah tak, jak byl vytvořen a uložen vývojářem.

2.4.1 PHP

PHP byla původně zkratka pro Personal Home Page, nyní se používá spíše Hypertext Preprocessor. Je to skriptovací programovací jazyk, který se využívá hlavně při tvorbě dynamických internetových stránek. PHP patří mezi dva nejrozšířenější skriptovací jazyky pro web. Je začleňován nejčastěji přímo do struktury jazyka HTML. Díky jednoduchosti použití, rozsáhle zásobě funkcí a kombinaci vlastností více programovacích jazyků se stal velmi oblíbeným (Procházka, 2012).

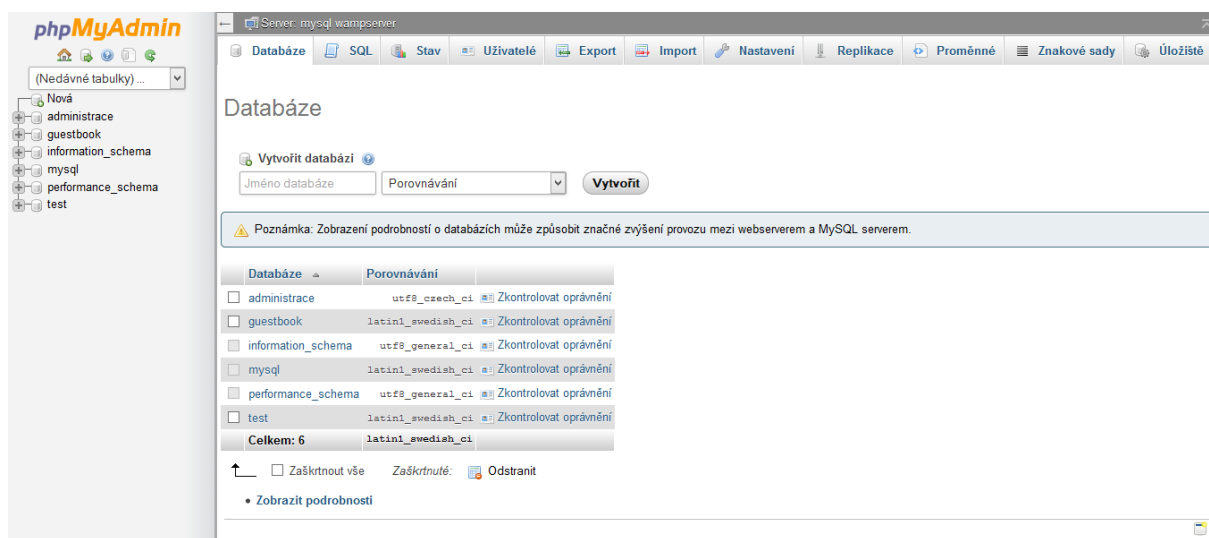
Historie PHP se datuje někdy kolem roku 1995, kdy vydal Rasmus Lerdorf sadu skriptů pod názvem PHP/FI, které byly napsány v jazyce Perl. Tato sada skriptů se setkala s úspěchem a stala se velmi používanou. Z toho důvodu byla roku 1997 vydána sada skriptů PHP/FI verze 2, tentokrát už napsána v jazyce C. Pro vývoj komerčních aplikací tato verze Lerdorfovi nevyhovovala, a tak ji společně se Zeevem Suraskim začali kompletně předělávat a výsledek označili jako PHP verze 3. Už v zimě 1997 začala práce na přepisu jádra PHP z důvodu zvýšení výkonu pro složité aplikace. PHP verze 4 s tímto novým jádrem nazývaným Zend se objevilo po dvou letech v roce 2000. Následně v roce 2004 po vylepšení jádra a doplnění dalších možností, byla vydána PHP verze 5. Na této verzi bylo největší revolucí vylepšení možností objektově orientovaného programování (Ponkrác, 2007).

Již v roce 2005 se objevili první zmínky o verzi 6. Ta se dlouhou dobu vyvíjela, ale neustále narážela na problémy. Proto bylo postupně vydáno šest verzí PHP 5, z nichž poslední verze PHP 5.6 byla vydána v srpnu 2014. Již v červenci roku 2014 bylo hlasováním rozhodnuto, že PHP verze 6 nebude vydána a další verze po verzích PHP 5.x, bude mít název PHP 7 (Lemos, 2014).

2.4.2 MySQL

Před příchodem počítačů byly databáze v papírové podobě a šlo o místa, kde byly uloženy nejruznější údaje, například informace o zaměstnancích, pacientech a podobně. Po příchodu počítačů se postupně všechny tyto informace začaly přesouvat do elektronické podoby. Z hlediska způsobů ukládání dat a vazeb mezi nimi rozlišujeme několik druhů databází. MySQL patří mezi relační databáze a je složená z tabulek, které slouží k uložení dat do paměti databáze. Tabulky obsahují sloupce a řádky do kterých jsou vkládány data.

MySQL je multiplatformní databáze, podporující dotazování pomocí jazyka SQL. I když není MySQL jedinou databází, prosadila se téměř jako nepsaný standard a stala se nejpoužívanějším databázovým systémem v rámci programování dynamických webových stránek. Výhodou této databáze je, že jí nabízí většina webhostingů které nabízí PHP a to včetně bezplatných webhostingů. Databázi MySQL lze provozovat na lokálním počítači prostřednictvím určitého databázového serveru a jednoduše tak simulovat webové prostředí (Procházka, 2012).



Obr. 2.17 – Správa MySQL databáze v phpMyAdmin

Souvislost mezi MySQL a PHP je v tom, že skript PHP může vystupovat jako aplikace v roli klienta databázového serveru. Je tedy možné se pomocí skriptu PHP spojit s databázovým serverem a využívat jeho služby. Skripty PHP mohou obsahovat příkazy v jazyce SQL a je možné databázový server požádat o práci s daty (Ponkrác, 2007).

2.4.3 JavaScript

JavaScript je jak již z názvu vyplývá skriptovací jazyk, zároveň se jedná o multiplatformní a objektově orientovaný jazyk. Je používán pro webové stránky a často je vkládán přímo do HTML kódu stránky. JavaScript slouží k tvorbě různých prvků pro oživení webu například animace, efekty obrázků, blikající texty. Umožňuje například hodnotit data ve formuláři, vytvoření hodin, jsou jím ovládány interaktivní prvky jako textové pole a tlačítka. JavaScript má s programovacím jazykem Java podobnou syntaxi, ale slovo Java v názvu je pouze z marketingových důvodů. Co se týče syntaxe, je JavaScript nejvíce podobný skriptovacímu programovacímu jazyku PHP. (Procházka, 2012).

Skript, neboli vše co je v JavaScriptu vytvořené je možno umístit přímo do stránky HTML dokumentu a to do sekce body nebo do sekce head podle účelu skriptu. Další možnost je vytvoření odkazu na daný skript, v takovém případě si jej stránka sama natáhne. Skript se zapisuje do stránky HTML mezi párový tag `<script>`. JavaScript je Case Sensitive, což znamená, že rozlišuje mezi velkými a malými písmeny.

```
<!DOCTYPE html>
<html lang='cs'>
  <head>
    <title>JavaScript</title>
    <script language="JavaScript" type="text/javascript">
      document.write("Tělo JavaScriptu");           //skript v hlavičce dokumentu
    </script>
  </head>
  <body>
    <script language="JavaScript" type="text/javascript">
      document.write("Tělo JavaScriptu");           //skript v těle dokumentu
    </script>

    <script language="JavaScript" type="text/javascript" src="externi_skript.js">
    </script>                                     <!-- externí skript -->

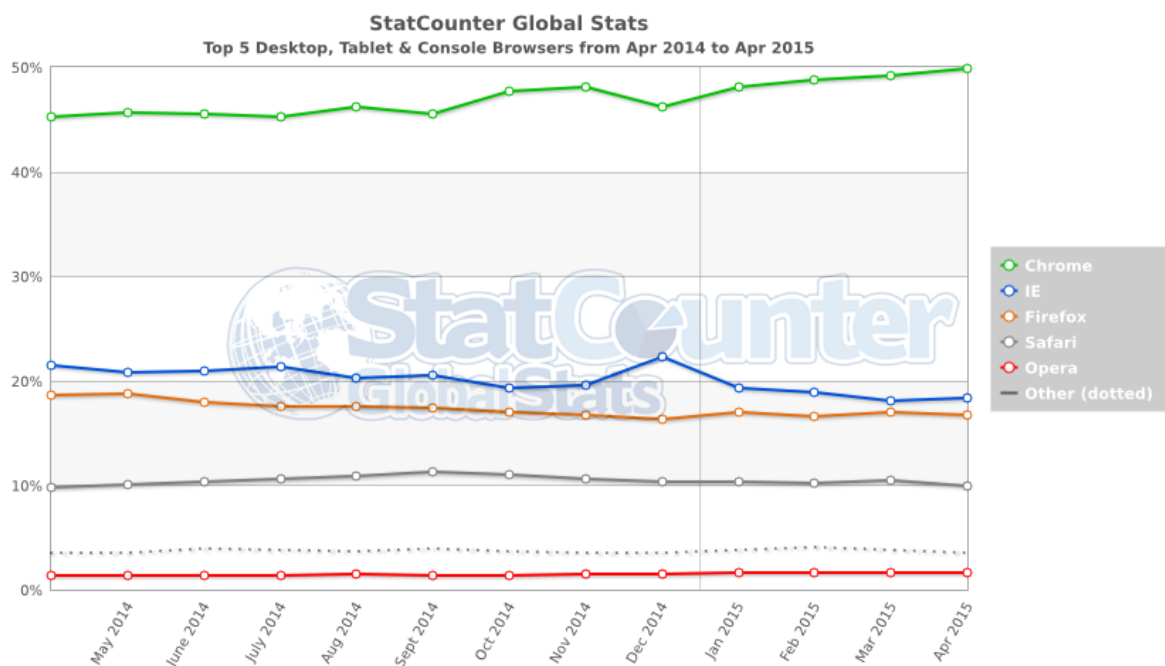
  </body>
</html>
```

Obr. 2.18 – Umístění skriptu v HTML dokumentu (Zdroj: autor)

2.5 Webové prohlížeče

Webový prohlížeč je program, který slouží k prohlížení webových stránek. V současné době existuje spousta zařízení, které obsahují webový prohlížeč. Na rozdíl od minulých let kdy bylo možné připojit se k internetu a dostat se na webové stránky pouze ze stolních počítačů případně notebooku, dnes mají vývojáři prohlížečů mnohem větší práci. Kromě stolních počítačů a notebooku je webový prohlížeč obsažen nejčastěji v mobilních telefonech, tabletech, herních konzolích a chytrých televizích.

To nese také řadu komplikací při tvorbě webových stránek. Ne všechny prohlížeče podporují nejnovější standardy a často dochází k odlišnostem při zobrazování obsahu webu. Dalším problémem jsou také starší verze prohlížečů, které neznají nové standardy. Web proto musí být vyvíjen tak, aby byl optimální pro co nejvíce prohlížečů. V některých případech dochází k používání prefixů. Například pro Internet Explorer verze 9 a nižší se používá JavaScript, který pomáhá těmto starším prohlížečům zobrazit nové prvky standardu HTML5.



Graf 2.1 – Roční srovnání nejoblíbenějších prohlížečů stolních počítačů, tabletu a konzolí (zdroj: <http://gs.statcounter.com/#browser-ww-monthly-201404-201504>)

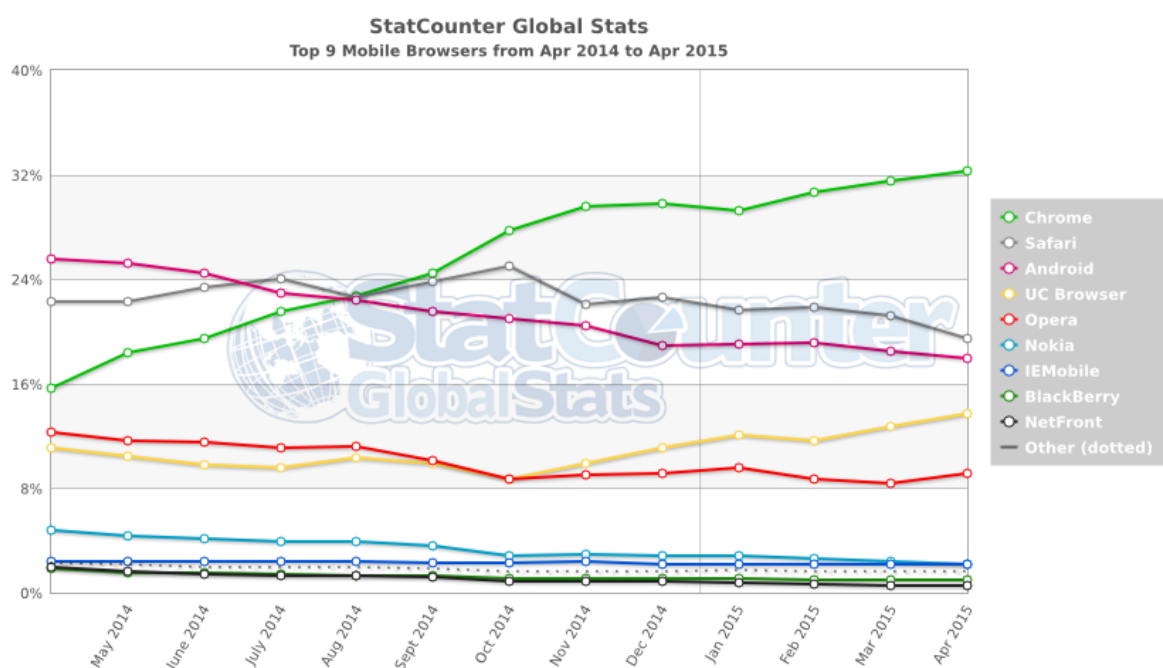
Z grafu 2.1 od serveru StatCounter.com vyplývá, že nejoblíbenější prohlížeč uživatelů stolních počítačů, tabletu a konzolí je Chrome od společnosti Google. Tento prohlížeč byl vydán v září roku 2008 a necelé čtyři roky poté, v roce 2012, se již stal nejoblíbenějším prohlížečem. Chrome tak sesadil z první příčky téměř dvacet let vládnoucí prohlížeč Internet Explorer. Jako největší výhody prohlížeče Chrome oproti jiným prohlížečům jsou udávány hlavně rychlost, jednoduchost, účinný marketing, vždy aktuální verze bez zbytečných reklamních kampaní a také pokročilé funkce pro vývojáře a webdesignery (Kasík, 2012).

Daleko pod prohlížečem Chrome skončil s oblíbeností kolem 18% uživatelů prohlížeč Internet Explorer od firmy Microsoft. Ten byl dlouhou dobu nejpoužívanějším prohlížečem na světě a to především proto, že si uživatelé neměli z čeho vybírat. S příchodem nových prohlížečů, především Google Chromu, jeho oblíba začala slábnout. Tento prohlížeč je druhým nejpoužívanějším především proto, že jej firma Microsoft zabudovává do každého operačního

systému Windows. Aby se firma Microsoft vyrovnala své konkurenci, nově nahrazuje prohlížeč Internet Explorer novým prohlížečem s názvem Microsoft Edge. Ten bude dostupný jako defaultní prohlížeč na všech zařízeních s operačním systémem Windows 10.

Těsně pod prohlížečem Internet Explorer se umístil prohlížeč Firefox od společnosti Mozilla. První verze tohoto prohlížeče byla vydána roku 2004 a stala se velmi oblíbenou. Firefox se stal dobrou konkurencí prohlížeče Internet Explorer, ale nikdy jej nepředstihl. Od roku 2010 oblíbenost tohoto prohlížeče mírně, ale stabilně klesá.

Další známe, ale už méně používané prohlížeče jsou Safari od společnosti Apple a Opera.



Graf 2.2 – Roční srovnání nejoblíbenějších prohlížečů v mobilních telefonech (Zdroj: http://gs.statcounter.com/#mobile_browser-ww-monthly-201404-201504)

Z grafu 2.2 vyplývá, že nejpoužívanějším prohlížečem u mobilních telefonů je tak jako u stolních počítačů prohlížeč Chrome od společnosti Google. Google Chrome je dostupný pro mobilní telefony s operačním systémem Android nebo iOS. Operační systém Android má sice svůj základní prohlížeč, ale jeho obliba klesá a v současnosti je na třetím místě hned za prohlížečem Safari od Applu. Operační systém Windows Phone OS používá obdobu desktopového prohlížeče Internet Explorer IEMobile, ten skončil až na sedmém místě.

2.6 HTML editory

Vývojáři mají k vytvoření webových stránek několik možností. Nejzákladnější z možností je tvorba stránek v jakémkoli textovém editoru a výsledný soubor uložit s příponou html nebo php. Tento způsob tvorby webových stránek je ale příliš zdlouhavý a nepohodlný, protože vývojář nemá k dispozici žádné pomůcky. Z těchto důvodů se používají HTML editory, které napomáhají vývojáři při práci s kódem. Celková kvalita webu ale není na HTML editorech závislá. Editory umožňují například automatické dopisování prvků, přístup k nápovědě jednotlivých prvků, odlišení barvy písma pro lepší rozeznání částí kódu a podobně.

HTML editory se dělí na základní dva typy, a to na strukturní, neboli ruční editory a na WYSIWYG, neboli vizuální editory.

2.6.1 Strukturní editory

Tyto editory umožňují tvorbu webových stránek pomocí ručního zápisu kódu. Strukturní editory ale nabízí oproti normálním textovým editorům spoustu vylepšení a nástrojů, které vývojáři ulehčí práci. Jde například o šablony, klávesové zkratky, zvýraznění syntaxe, automatické opravy a spoustu dalších funkcí. Mezi tyto editory patří například PSPad, Notepad++, NetBeans, nebo SharePoint Designer.

2.6.2 WYSIWYG editory

Zkratka WYSIWYG je z anglického spojení „What You See Is What You Get“, což v českém jazyce znamená „Co vidíš, to dostaneš“. U těchto vizuálních HTML editorů je tvorba webové stránky poměrně jednoduchá. Editor veškerý kód vygeneruje za vývojáře sám. Vývojář vytváří prvky stránky jako v normálním editoru, jakým je například Microsoft Word, aniž by musel použít HTML kód. Častým problémem těchto editorů byl zmatek ve výsledném kódu. Postupně proto WYSIWYG editory začaly nabízet vývojářům možnost nahlédnout do kódu a provést úpravy. Mezi WYSIWYG editory patří například Microsoft Frontpage, Adobe Dreamweaver, NicEdit nebo TinyMCE.

2.6.3 Redakční systémy

Webové redakční systémy vypomáhají s vytvářením a správou webu a jeho obsahu. Tyto systémy jsou určeny pro ty, kteří téměř nebo vůbec neznají HTML a CSS kód a chtějí se starat o svůj web a často měnit obsah. Především usnadňují vytváření, kontrolu a editování obsahu webu. Mezi redakční systémy patří například WordPress, Joomla!, nebo Drupal.

3. Koncepce webové prezentace

3.1 Produkční plán

Před zahájením tvorby samotných webových stránek je vhodné zvolit postup, jakým způsobem by měly být stránky vytvářeny. K tomuto účelu slouží produkční plán, který se skládá z několika bodů, které charakterizují budoucí záměry, plány a cíle.

3.1.1 Název webu

Volba názvu webu je v tomto případě trochu složitější, než by se mohlo zdát. Jde o to, že webové stránky jsou vyvíjeny pro kadeřnické studio, které nemá svůj jednoznačný název. V kadeřnictví působí pravidelně jedna kadeřnice v pánském oddělení a jedna kadeřnice v dámském oddělení. Přitom v pánském oddělení některé dny nebo části dnů zaskakuje třetí kadeřnice. Nabízí se tedy varianty pojmenovat web podle jména jedné z kadeřnic, či spojení jmen dvou hlavních kadeřnic a poslední varianta je nazvat web podle města, ve kterém se kadeřnictví nachází. Po dohodě byla vybrána varianta spojení křestních jmen dvou hlavních kadeřnic Dagmar a Alžběty. V případě Alžběty byl zvolen na vyžádání slangový název jména, Bětka. Jelikož by doména u této varianty byla velmi dlouhá, bylo rozhodnuto zkrátit ji a použít začáteční písmena obou jmen, tedy d a b. Samostatné dab.cz je ale nic neříkající a proto bylo doplněno o klíčové slovo kadeřnictví. Doména byla tedy zvolena na kadernictvidab.cz.

3.1.2 Záměr a cíle

Hlavním cílem této webové prezentace je zviditelnit kadeřnictví, které doposud nemělo svou vlastní webovou prezentaci, což je v dnešní době nepřípustné. Mezi další záměry patří samozřejmě přilákání nových zákazníků a větší informovanost zákazníku, kteří budou moci vidět změny v otvírací době, případně aktualizace ceníku z pohodlí domova. V neposlední řadě bylo cílem vyrovnat se konkurenci ve městě, která svou webovou prezentaci již má.

3.1.3 Cílové publikum

Cílové publikum jsou potencionální zákazníci. Jedná se o lidi ve věkovém rozmezí od patnácti do čtyřiceti let, kteří jsou více ochotní změnit svého kadeřníka, než starší lidé. Další kategorií jsou zákazníci, kteří bydlí v okolních vesnicích nedaleko města. Poslední zaměřeným publikem jsou stálí zákazníci a jejich větší informovanost.

3.1.4 Vzory návštěvnosti

Předpokladem je, že uživatelé přicházejí na web přes vyhledávače, kdy vyhledávají přímo kadeřnictví v okolí, nebo na web narazí při hledání informací o městě. Menší část uživatelů přichází na web přímo zadáním adresy webu, tyto uživatelé se o webu dozvědí od známých, případně z oběžníku. Většina uživatelů přichází na web jen jednou případně dvakrát a po zjištění potřebných informací už nemají důvod se vracet. Uživatelé na webu stráví přibližně pět až deset minut.

3.1.5. Příjmový model

Webová prezentace je tvořena ke zviditelnění kadeřnictví a přilákání nových zákazníků. S příchodem nových zákazníků je samozřejmě spojeno zvýšení zisku. Do budoucna by bylo možno zvýšit příjmy pomocí reklam zaměřených na daný obor.

3.1.6 Sada schopností

Webová prezentace byla složena tak aby se v ní mohl uživatel pohybovat co nejsnadněji a nejpřehledněji. Hned pod hlavičkou stránky je umístěno horizontální navigační menu, které je složeno z pěti položek. První z nich je úvod, kde uživatelé naleznou nejzákladnější informace. Dále následuje ceník služeb, rozdělený podle pánských a dámských služeb. Třetí položka v pořadí je galerie, kde si mohou uživatelé prohlédnout fotografie interiéru a exteriéru. Následuje kniha návštěv, která umožňuje uživatelům napsat své připomínky či pochvaly a doporučení. Poslední položkou jsou kontakty, kde jsou uvedeny jména a kontakty jednotlivých kadeřnic a mapa s adresou kadeřnictví. Po celou dobu mají uživatelé k dispozici otevírací dobu, která byla umístěna do levého sloupce stránky. Celý web zakončuje zápatí, kde jsou umístěny informace o autorovi a validacích kódu. Žádné reklamy z počátku na webu nebudou umístěny. Vyhledávací nástroje rovněž nebudou zapotřebí, jelikož je tento web malou, jednoduchou a přehlednou prezentací.

3.1.7 Konkurence

Kadeřnictví se nachází v malém městě Šenov, které leží mezi Ostravou a Havířovem. Konkurencí jsou samozřejmě kadeřnictví v těchto dvou velkých městech, protože se nenacházejí až tak daleko. Největší konkurencí jsou ale kadeřnictví přímo v Šenově a blízkých přilehlých obcích. V Šenově se nachází dvě konkurenční kadeřnictví, která ale nemají tak dlouhou tradici a byly zřízeny teprve v nedávných letech.

Seznam konkurence:

- Kadeřnictví Jana – Šenov – web: <http://kadernictvijanule.webnode.cz/>,
- X-Tra Hair Studio Veronique – Šenov – web: bez webové prezentace,
- Studio Gabriela – Petřvald – web: <http://www.studiogabriela.cz/>,
- Petra Hlůžová – Ostrava Radvanice – web: bez webové prezentace.

3.1.8 Marketingový plán

Webová prezentace byla vytvořena především ke zviditelnění kadeřnictví a přilákání nových zákazníků. Dlouhodobým cílem bude dostat web do předních míst nejznámějších vyhledávačů tedy Googlu a českého Seznamu. Pro začátek bude web umístěn do katalogů těchto vyhledávačů. Další možností je využití dnes velmi populárních sociálních sítí k prezentaci webu pro širokou veřejnost. K zajištění webu bude využito otištění adresy webu do místního oběžníku.

3.1.9 Tým

Webová prezentace nemusí být často aktualizována. Jedinými změnami mohou být změny v ceníku služeb či otevírací doby, nebo obměna fotografií. Na správu tohoto webu postačí jedna osoba.

3.1.10 Předpoklady

Předpokladem tohoto webu je informovat případné potencionální zákazníky o existenci kadeřnictví, které doposud nebylo nijak propagováno. Také zajištění informací pro stávající zákazníky a možnost vyjádřit se spokojeností případně nespokojeností s nabízenými službami. Tyto aspekty pomohou s rozvojem a zlepšováním poskytovaných služeb.

3.2 Základní části webu

Celý web je uzavřen do obalu, který umožňuje vycentrovat stránku na střed monitoru. Dále je webová prezentace rozdělena do šesti základních částí, které umožňují uživatelům snadno se orientovat.

3.2.1 Záhloví

Záhloví webu je pouze informativní a neobsahuje žádné funkcionality ani odkazy. Záhloví webu informuje uživatele o názvu webové prezentace prostřednictvím loga. Logo bylo vytvořeno spojením křestních jmen dvou kadeřnic, jelikož kadeřnictví nemá svůj pevně daný název. Dále bylo logo obohaceno o klíčové slovo kadeřnictví a tematický vektorový obrázek v podobě kadeřnických nůžek a hřebenu. K tvorbě byl použit vektorový grafický program Corel Draw. Logo má transparentní pozadí a je uloženo ve vektorovém formátu svg (Scalable Vector

Graphics). Výhoda tohoto formátu spočívá ve snadné měnitelnosti velikosti obrázků, aniž by se měnila velikost samotného souboru.



Obr. 3.3 – Logo

Dále je v záhlaví webu umístěn stylový ornament a silueta obličeje ve webových barvách, rovněž ve formátu svg. Tyto obrázky doplňují záhlaví a vytváří lepší vizuální pohled na web.

3.2.2 Horizontální navigační menu

Horizontální navigační menu je situováno těsně pod záhlavím webu. Obsahuje pět odkazů, pomocí kterých se uživatelé dozvědí nejdůležitější informace o kadeřnictví. Prvním odkazem, umístěným nalevo, je úvod, kterým jsou uživatelé přivítáni na webu a kde lze nalézt základní informace o kadeřnictví a jeho zaměstnancích. Následujícím odkazem je ceník. Ceník služeb je rozdělen do dvou částí na pánské a dámské oddělení. Dalším odkazem v pořadí horizontálního navigačního menu je galerie. Galerie je rovněž rozdělena na pánské a dámské oddělení, navíc je zde i kategorie exteriér kadeřnictví, kde uživatele mohou vidět například parkoviště u kadeřnictví. Jako galerie je použit lightbox. Lightbox je javascriptová aplikace která slouží k efektivnímu zobrazení fotografií či obrázků. Po kliknutí na fotografii se zatmaví okno prohlížeče a efektně se zobrazí fotografie ve větším rozlišení. Uživatel následně může pomocí šipek zobrazovat přímo další fotografie v pořadí v dané kategorii. Čtvrtým odkazem je kniha návštěv. Pomocí knihy návštěv mohou uživatelé vyjádřit své dojmy z návštěvy kadeřnictví, napsat své připomínky, případně se na něco zeptat. Posledním odkazem v horizontálním menu webu jsou kontakty. Zde mohou uživatelé nalézt kontakty na zaměstnance, identifikační čísla a adresu provozovny. Pod těmito informacemi je umístěna mapa od společnosti Google, na které je vyobrazen bod se sídlem kadeřnictví. S mapou mohou uživatelé dále pracovat, kromě přiblížení a oddálení mapy je možné například naplánovat trasu.

3.2.3 Levý sloupec

V levém sloupci webu je umístěna otevírací doba. Ta je rovněž rozdělena na pánské a dámské oddělení. Uživatelé tak mají k dispozici po celou dobu jejich přítomnosti na webu otevírací dobu.

3.2.4 Hlavní část

Hlavní část webu neboli pravý sloupec, je umístěna vedle levého sloupce pod horizontálním navigačním menu. V hlavní části webu je zobrazován veškerý obsah odkazů umístěných v navigačním menu.

3.2.5 Zápatí

Poslední částí webu je zápatí, které bývá často slangově označeno jako patička webu. Zápatí slouží k uvedení informací o autorovi webu, ale i mnoho dalších užitečných informací. V tomto případě je v zápatí uvedená také mapa webu. Mapa webu slouží pro lepší orientaci uživatelů na webu. Umožňuje uživatelům vidět celou strukturu stránek naráz, což může napomoci uživatelům, kteří se ztratili, nebo kteří se na webu špatně orientují. Dále jsou v zápatí umístěny ikony w3c organizace dokazující validní HTML a CSS kód a ikona SEO servisu odkazující na výsledek analýzy zdrojového kódu webu.

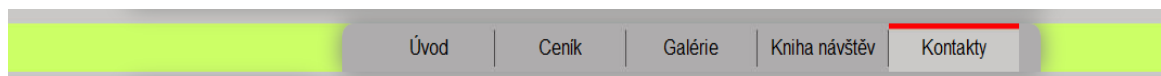
3.3 Grafická podoba webu

Při zadání požadavků na zpracování webové prezentace nebyly zadavatelem specifikovány žádné speciální grafické požadavky. Byla požadována přehlednost a jednoduchost webových stránek, vzhledem k očekávání, že uživatelé webu budou i méně zdatní ve využívání informačních technologií. Pro web byly zvoleny dvě základní barvy různých odstínů, kterými jsou světle zelená a šedá, jak lze vidět na obrázku 3.4. Barva písma je na většině webu černá.



Obr. 3.4 – Úvodní strana webu (Zdroj: autor)

Jedinou odlišnou barvou byl zvolen malý červený pruh, který se zobrazí při najetí kurzorem na odkaz, jak lze vidět na obrázku 3.5. Tento prvek dostatečně odliší vybraný odkaz a napomáhá k zlepšení celkové přehlednosti webu.



Obr. 3.5 – Zvýraznění odkazu (Zdroj: autor)

3.4 HTML kód

Při tvorbě webové prezentace byl využit značkovací jazyk HTML ve své nejnovější verzi 5. Snahou bylo využívat těchto nových prvků a nahradit jimi zastaralé způsoby. S tím přichází samozřejmě komplikace se zobrazením ve starších prohlížečích. Největší problémy jsou pochopitelně s prohlížečem Internet Explorer, jehož starší verze využívá stále hodně lidí. Už jen proto, že tento prohlížeč byl integrován v operačních systémech Windows dlouhá léta. Tento problém byl vyřešen Remy Sharpem, který vytvořil skript zpřístupňující HTML5 pro prohlížeče Internet Explorer verze 9 a nižší. Tento skript má otevřený zdrojový kód a lze jej použít v jakémkoli projektu. Základní myšlenka skriptu lze vidět na obrázku 3.6.

```
1 <!--[if lt IE 9]>
2 <script>
3   var e = ("abbr,article,aside,audio,canvas,datalist,details," +
4     "figure,footer,header,hgroup,mark,menu,meter,nav,output," +
5     "progress,section,time,video").split(',');
6   for (var i = 0; i < e.length; i++) {
7     document.createElement(e[i]);
8   }
9 </script>
10 <![endif]>
```

Obr. 3.6 – Základní myšlenka zpřístupňujícího skriptu (Zdroj: autor)

Skript je uložen v podmíněném komentáři. Ten Internet Explorer interpretuje jako příkaz if, jestliže je aktuální prohlížeč Internet Explorer verze nižší než 9 provede následující blok. Ostatní prohlížeče berou celý blok jako HTML komentář, tudíž je tento blok ignorován a stránka se načte rychleji. Skript musí být umístěn na začátku stránky, nejlépe v sekci head aby jej mohl Internet Explorer provést před analýzou všech značek a atributů.

Remy Sharp skript umístil na Google Project Hosting, to umožňuje skript vložit do webu pomocí odkázání na tuto hostovanou verzi, jak lze vidět na obrázku 3.7.

```
1 <!--[if lt IE 9]>
2   <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
3 <![endif]-->
```

Obr. 3.7 – Zkrácený zápis pomocí odkazu (Zdroj: autor)

Po implementaci tohoto skriptu by měli i starší verze prohlížeče Internet Exploreru umět zacházet s novými prvky značkovacího jazyka HTML verze 5.

V tomto projektu byli použity prvky HTML5, těmi nejzákladnějšími jsou prvky header, section, nav, aside, article a footer.

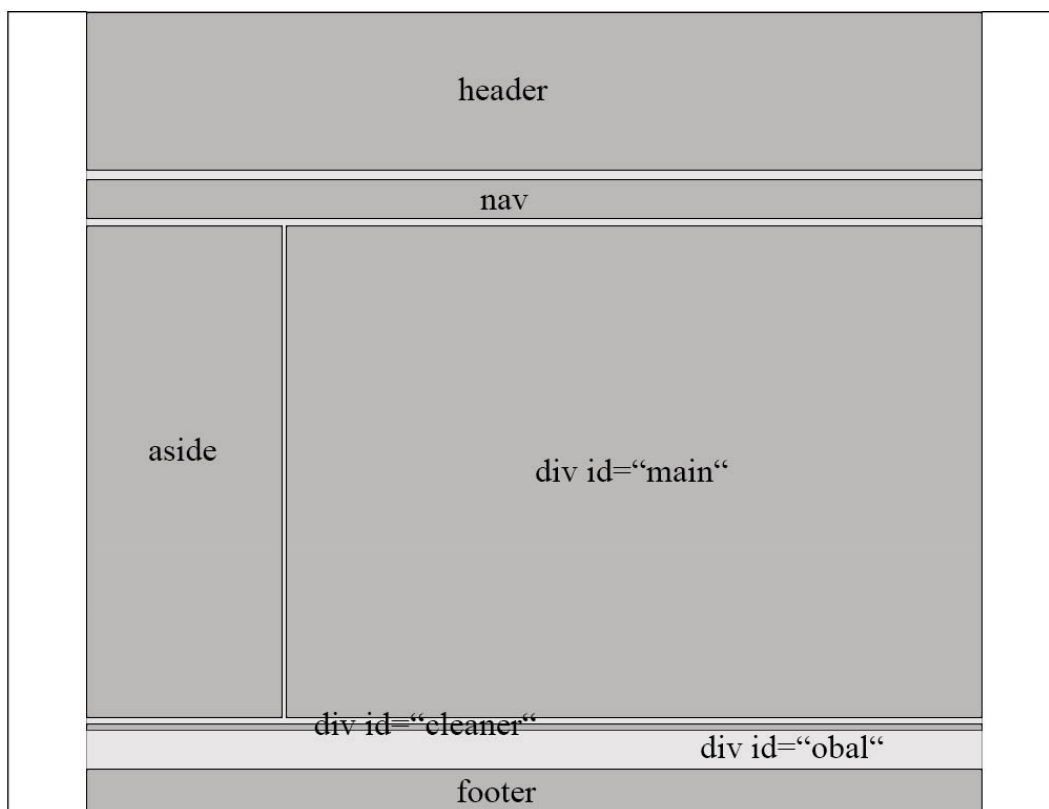
```
13 <body>
14   <div id="obal">
15     <header></header>
16     <nav></nav>
17     <div id="main">
18       <article>
19         <section>
20         </section>
21       </article>
22     </div>
23     <aside></aside>
24     <div id="cleaner"></div>
25     <footer></footer>
26   </div>
27 </body>
```

Obr. 3.8 – Základní struktura těla úvodní stránky v HTML kódu (Zdroj: autor)

Pro záhlaví kódu, které se v minulosti řešilo pomocí prvku div s jednoznačným identifikátorem, byl pomocí HTML5 využit prvek header. Další nové prvky v záhlaví nebyly použity z důvodu, že se zde nachází pouze logo a doplňující obrázky. Pro horizontální navigační menu byl použit prvek nav. Levý sloupec, ve kterém je k dispozici otevírací doba, je řešen pomocí prvku aside, který je vhodný pro obsah který je oddělený od hlavního obsahu stránky. Samotný hlavní obsah je zabalen do prvku div s identifikátorem main. Žádný nový prvek se pro tuto část v téhle práci nehodil. Hlavní část a levý sloupec odděluje od zbytku stránky čistič vytvořený pomocí prvku div umístěný do třídy cleaner. Tento oddělovač je zde umístěn, aby bylo ukončeno obtékání a zápatí stránky bylo vždy dole. Celou stránku ukončuje zápatí, na které byl použit prvek footer.

Následně byly tyto části pomocí kaskádových stylů pozicovány do požadované podoby. Rozvržení jednotlivých popisovaných prvků lze vidět na obrázku 3.10.

Mezi další nové prvky HTML5 které byly použity, patří article. Ten slouží k zabalení komentářů, které uživatelé vkládají prostřednictvím knihy návštěv. Prvek article byl použit také při tvorbě úvodní stránky webu, kde slouží k zabalení úvodního článku. S tím souvisí také prvek hgroup. Ten byl použit pro sdružení dvou nadpisu v úvodním článku.



Obr. 3.9 – Rozvržení prvků webu (Zdroj: autor)

3.5 CSS kód

Tak jako tomu bylo u HTML také u CSS kódu bylo snahou využívat nejnovější funkce CSS verze 3. Celý web je uložen v blokovém elementu div s identifikátorem obal. Tomuto obalu byla pomocí kaskádových stylů nastavena šířka na 850 pixelů a následně byl pomocí vlastnosti margin a hodnoty 0 auto vycentrován. Obalu byla nastavena také minimální délka, a to na 100 procent. Je to jedna z deklarací, která zajistí, aby se zápatí webu drželo vždy u spodní části obrazovky či prohlížeče, i v případě, že je na webu málo obsahu.

```

16 #obal
17 {
18     width: 850px;
19     margin: 0 auto;
20     min-height: 100%;
21     position: relative;
22 }

```

Obr. 3.10 – Deklarace pro obal (Zdroj: autor)

K zabránění častého používání jednoznačných identifikátorů a tříd byly využívány i speciální selektory typu následník, sourozenec, sdružování selektoru či pseudotřídy, například first-child a last-child. Konkrétně pseudotřída last-child patří mezi novinky, které přišli s novou verzí CSS 3.

```

250 .cenik td:first-child
251 {
252     padding-right: 80px;
253     border-bottom: 1px solid #1eb00a;
254 }
255 .cenik td:last-child
256 {
257     text-align: right;
258     border-bottom: 1px solid #818181;
259 }
260 .cenik + p
261 {
262     text-align: center;
263     margin: 5px;

```

Obr. 3.11 – Pseudotřída a selektor sourozenec (Zdroj: autor)

Na obrázku 3.11 lze vidět využití těchto selektorů a pseudotříd. Řádek 250 je určen pro buňku, která je prvním potomkem nějakého jiného prvku, v tomhle případě tedy řádku tabulky. Následně má nastaven pravý vnitřní okraj na 80 pixelů a barevné podtržení. Řádek 255 vyjadřuje přesný opak, tedy buňka, která je posledním potomkem nějakého jiného prvku, v tomhle případě řádku tabulky. Následné deklarace vyjadřují vlastnost zarovnání textu s hodnotou vpravo a rovněž barevné podtržení, vytvořeno pomocí dolního rámečku. Na řádku 260 lze vidět typový selektor sourozenec, který znamená prvek p bezprostředně předcházený třídou ceník. Pravidlo tedy platí pro první odstavec za třídou ceník.

Celý web budí dojem, že je poskládán z obdélníků a čtverců se zaoblenými rohy. Tato vlastnost s názvem border-radius je rovněž novinkou, která přišla s třetí verzí kaskádových stylů, stejně jako vlastnost box-shadow, pomocí které jsou vytvářeny stíny těchto objektů.

```

24 header
25 {
26     border-bottom-left-radius: 15px;
27     border-bottom-right-radius: 15px;
28     box-shadow: 0px 0px 20px #808080;
29 }

```


Obr. 3.12 – Použití zaoblení rohů a stínů (Zdroj: autor)

Zaoblení rohů může být aplikováno na libovolný počet rohů, jak lze vidět v obrázku 3.12. V tomto projektu bylo použito nejčastěji zaoblení na všechny čtyři rohy, či horní nebo dolní zaoblení. Vlastnost border-radius umožňuje také zakulacovat ve tvaru elipsy, a to přidáním lomítka a další hodnoty. V kódu lze zkráceně zapisovat pouze border-radius a následně 4 hodnoty od levého horního rohu ve směru hodinových ručiček.

Na obrázku 3.12 lze také vidět, jak byla v projektu deklarována vlastnost box-shadow. První hodnota udává horizontální a druhá vertikální posun směrem dolů. V tomto projektu byly tyto hodnoty nastaveny na 0, protože bylo žádoucí mít stín kolem celého objektu. Další hodnota v pořadí udává rozostření stínu. To bylo nastaveno na 20 pixelů, aby byl stín jemný a méně výrazný. Nakonec je nastavena barva stínu, kterou byla zvolena šedá. Vlastnost box-shadow má k dispozici ještě další hodnotu, která definuje roztažení do stran.

3.6 Kniha návštěv

Na webu byla vytvořena kniha návštěv, kde se mohou uživatelé podělit o dojmy z návštěvy kadeřnictví, napsat své připomínky či se na něco zeptat. Kniha návštěv je vytvořena v jazyku php. Pomocí služby phpMyAdmin byla vytvořena databáze guestbook a v ní tabulka comments, která obsahuje pět záznamů, strukturu tabulky lze vidět na obrázku 3.13.

#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Další
1	id 	int(11)			Ne	Žádná	AUTO_INCREMENT
2	name	varchar(255)	utf8_general_ci		Ne	Žádná	
3	email	varchar(255)	utf8_general_ci		Ne	Žádná	
4	message	text	utf8_general_ci		Ne	Žádná	
5	timestamp	int(255)			Ne	Žádná	

Obr. 3.13 – Struktura tabulky comments v phpMyAdmin (Zdroj: autor)

Uživatel musí zadat jméno, email a zprávu, pokud některé z těchto polí nevyplní, objeví se chybová stránka, kde bude upozorněn na to, že musí vyplnit všechna pole. Textové pole pro vložení zprávy je omezeno na maximální délku 400 znaků. Samotný formulář zpracovává soubor post.php. Spojení s databází je prováděno pomocí souboru config.php, kde je uloženo jméno hosta, uživatelské jméno, heslo a název databáze. Následně je provedena kontrola, zdali jsou všechna pole vyplněna. Jestliže je vše v pořádku v souboru nk.php je zobrazena zpráva od uživatele. Ta se skládá ze jména uživatele, data a času vložení a samotné zprávy.

4. Responzivní design a jeho implementace

4.1 Proč responzivní design

V posledních letech neuvěřitelně rychle přicházejí stále nová zařízení a nejenom to. Také webových prohlížečů je daleko víc a uživatelé si sami vybírají, na jakém zařízení, či v jakém prohlížeči chtějí web prohlížet. Stále více uživatelů v prohlížečích zvětšuje či zmenšuje velikost písma, nebo pracují v minimalizovaném okně prohlížeče. Někteří uživatelé používají nejnovější zařízení, jiní zase zůstávají u starších, na které jsou zvyklí. Tyto fakta vedou k zamyšlení, jestli by měl web vypadat stejně ve všech situacích, jak tomu bylo kdysi. I když se zdá, že u projektu webové prezentace pro kadeřnictví není responzivní design až tak důležitý, v dnešní době chytrých mobilních telefonů a tabletu je přinejmenším užitečný. Pro uživatele je určitě příjemnější, když najdou potřebné informace na svém zařízení přehledně, bez nutnosti měnit velikost zobrazení a následně hledat posouváním prstem.

4.2 Plánování

Před zahájením tvorby responzivního designu je třeba zvážit nejdůležitější faktory včetně otázky zdali responzivní design vůbec vytvářet. Na tuto otázku bylo odpovězeno již v předešlé části – responzivní design je v tomto projektu žádoucí využít. Dále je potřeba si promyslet kde se bude responzivní design testovat či jaké provést analýzy.

4.2.1 Faktory ke zvážení

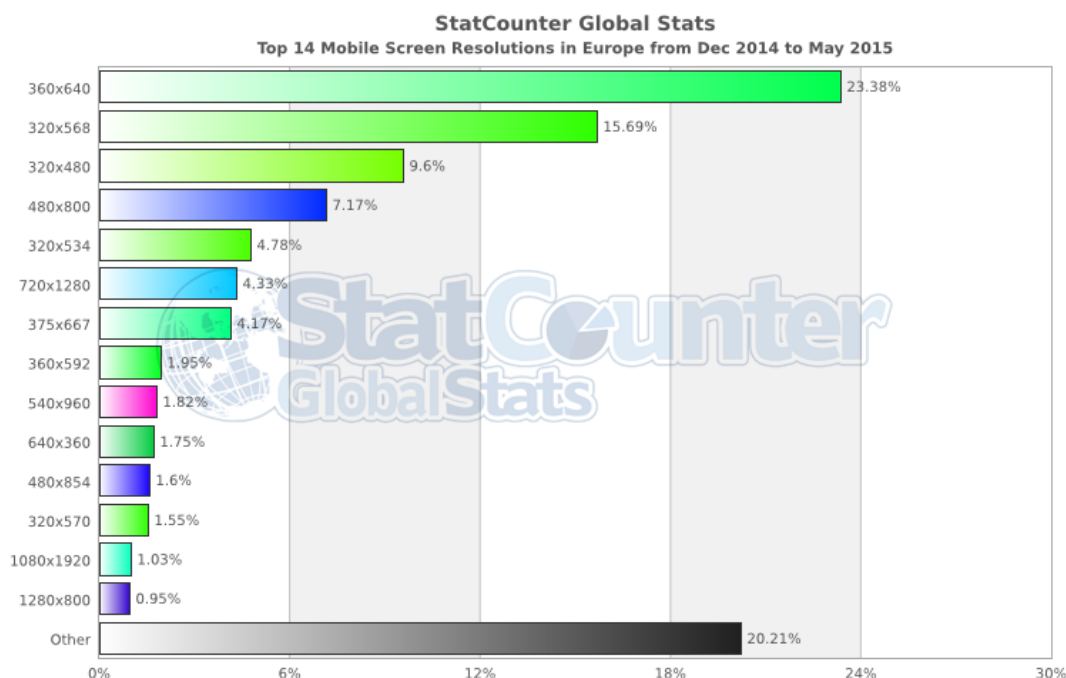
Jelikož se jedná o malý web, který slouží hlavně k prezentaci kadeřnictví, není až tolik faktorů ke zvažování. Co se týče výkonu, web neobsahuje mnoho externích skriptů či kaskádových stylů, které by se museli stahovat a zatěžovali by tak zařízení využívající mobilní síť. Časové hledisko v tomhle případě rovněž není takový problém, jelikož je možné responzivní design rozvíjet a vylepšovat i po zavedení webu a předání projektu. Výsledný design bude odzkoušen i na horších prohlížečích pro zajištění lepší podpory. Reklamy se na

webu nebudou vyskytovat. Maximálně reklama z webhostingu, či odkaz na dodavatele kadeřnických potřeb. Tyto reklamy budou statické a nebudou se obměňovat.

4.2.2 Analýzy

Do budoucna je možno použít některý z nástrojů jako je Google Analytics přímo na webu a sledovat, z jakých zařízení uživatelé web nejčastěji navštěvují, či jaké prohlížeče používají nejčastěji. Podle těchto dostupných informací následně optimalizovat web.

Je obecně známo, že přístupy na web z mobilních zařízení každoročně rostou. Podle statistiky společnosti StatCounter v Evropě využívá přes 22% uživatelů mobilní zařízení v přístupu na internet. V Africe je dokonce mobilní zařízení hlavní branou na internet.



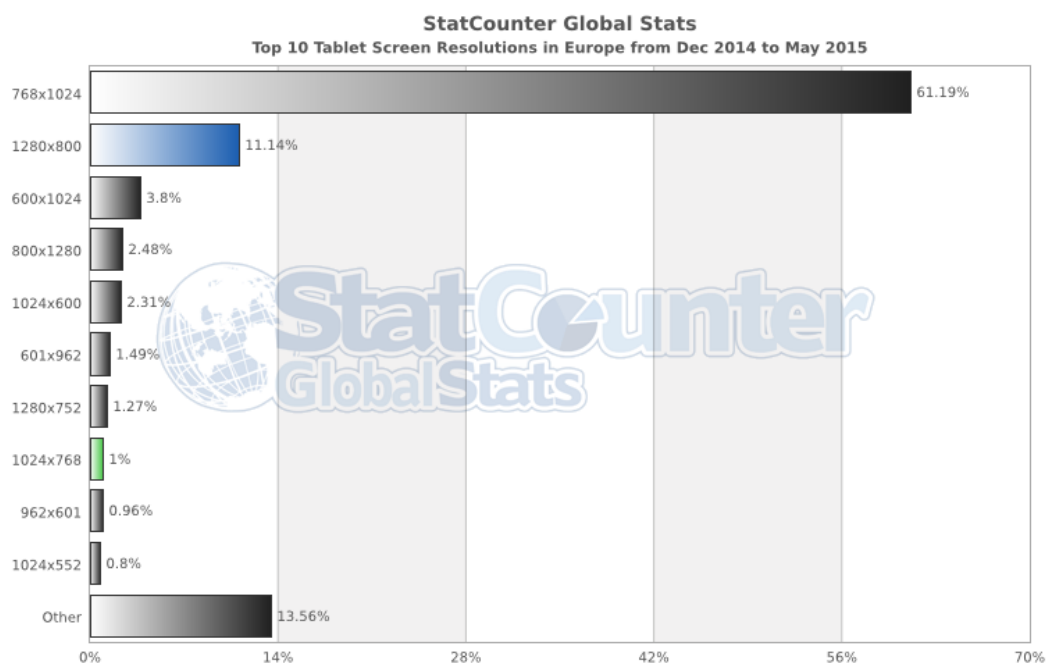
Obr. 4.1 – Top 7 mobilních rozlišení v Evropě (Zdroj:

http://gs.statcounter.com/#mobile_resolution-eu-monthly-201412-201505-bar)

Z obrázku 4.1 kde lze vidět čtrnáct nejrozšířenějších rozlišení mobilních telefonů v Evropě vyplývá, že vede rozlišení 360x640 pixelů, které využívá přes 23% mobilních telefonů. Proto bylo zvoleno toto rozlišení jako primární při budování responzivního designu. V grafu lze vidět, že dnes existuje opravdu velké množství rozlišení. Z toho důvodu je důležité nezaměřovat se pouze na optimalizaci pro určité zařízení, či rozlišení, ale snaha o to, aby byl web co nejvíce přístupný.

U tabletu má obrovskou převahu rozlišení 768x1024 pixelů, které využívá přes 61% tabletu v Evropě, jak vyplývá z grafu na obrázku 4.2. Z tohoto důvodu se toto rozlišení také stalo významnějším při budování responzivního designu.

Samozřejmě, že ne všechny analýzy jsou zcela přesné. Výsledky se mohou lišit a mohou být zkreslené v závislosti na geografické poloze, tamních sociálních podmínkách a podobně. Pro tento projekt postačí znát alespoň nejvyužívanější rozlišení a aplikovat základy responzivního designu.



Obr. 4.2 – Top 10 rozlišení tabletů v Evropě (Zdroj: <http://gs.statcounter.com/#tablet-resolution-eu-monthly-201412-201505-bar>)

4.2.3 Testovací prostředí

K testování funkčnosti responzivního designu byla použita skutečná zařízení, a to chytré mobilní telefony s velikostí displeje 5,5 palců, 5 palců a 4,7 palců. Tyto rozměry displeje jsou v současné době asi nejpoužívanějšími u chytrých mobilních telefonů. Pro testování responzivního designu na tabletech byl využit tablet s velikostí displeje 10 palců. Dalšími využitými testovacími zařízeními byl notebook s velikostí displeje 15,6 palců stolní počítač s monitorem o velikosti 22 palců. Na všech těchto zařízeních byly instalovány minimálně 2 prohlížeče k testování.

Kromě skutečných zařízení, což je nejlepší způsob testování, byly použity také emulátory a simulátory dostupné na internetu. Nevýhoda emulátoru je, že musejí být staženy na disk, kde

také zabírají hodně místa. Další nevýhodou emulátoru a simulátoru je, že nejsou příliš přesné a chyby, které se objeví na skutečném zařízení, se na těchto simulátorech objevit nemusí. Jedním z těchto nástrojů je <http://deviceponsive.com/> kde stačí zadat url adresu testovaného webu. Ten se následně zobrazí v 5 různých zařízeních a to na výšku i na šířku. Další využívaný web byl [web screenqueri.es](http://screenqueri.es) kde lze po registraci zdarma využívat beta verzi, která nabízí výběr ze spousty zařízení, možnost upravovat si velikost rozlišení a další užitečné funkce a informace. Poslední používaný web byl <http://ami.responsivedesign.is/>, který je spíše doplňkový a hodí se pro vytvoření screenshotů.

4.3 Jednotky

Základem responzivního designu je mít flexibilní layout. V tomto projektu byl zvolen plovoucí (fluid) layout. Plovoucí layout spočívá v tom, že se rozměry určují v procentech na místo pixelů. Po aplikaci tohoto layoutu je web schopen přizpůsobit svou šířku podle aktuální šířky prohlížeče. Nevznikají tak vodorovné posuvníky, či velké množství prázdného místa. V celém projektu jsou proto nejčastěji používána procenta, které jsou použity pro nastavení velikosti šířky objektů, vnitřních a vnějších okrajů, obrázků a podobně. Samozřejmě, že v některých případech muselo dojít ke kompromisu a proto je další používanou jednotkou pixel.

Velikost fontu byla nastavena pro celý dokument pomocí procent a to na 100%. U většiny prohlížečů je výchozí velikost fontu 16 pixelů. Toto nastavení je z důvodů, že některé prohlížeče nevěděly, jak hodně zmenšit či zvětšit písmo, pokud byla základní velikost fontu stránky nastavena podle jiných jednotek. Následně byly velikosti fontů ve zbytku webu upravovány pomocí flexibilní jednotky rem. Jednotka rem je stejná jako jednotka em, ale má výhodu, že se její velikost vztahuje ke kořenovému elementu, což zlehčuje práci a zvyšuje přehlednost.

4.4 Mediální dotazy

Plovoucí layout umožní, aby se web zobrazoval co nejlépe na velkém počtu rozlišení, zbylé záležitosti ohledně responzivního designu řeší mediální dotazy. Mediální dotazy umožní definovat podmínky, jako například rozlišení, výška a šířka zařízení, kdy se mají aplikovat speciální kaskádové styly.

4.4.1 Viewporty

Viewport je viditelná oblast, či šířka prohlížeče. U mobilních telefonů jsou ale viewporty komplikovanější. Mobilní telefony rozlišují mezi pixely zařízení a pixely CSS. Pixely zařízení jsou stále stejné a nemění se, zato pixely CSS se mohou měnit. Například na mobilních telefonech s vysokým rozlišením může být 1 pixel CSS roven dvěma pixelům zařízení. Uživatel může oddalovat, či přibližovat stránku a s tím se také mění velikost CSS pixelů. Z toho důvodu jsou viewporty rozděleny na viewport layoutu, který je pořád stejný a jeho míry se nemění a vizuální viewport, který se mění. Vizuální viewport je ta část stránky, kterou vidíme na obrazovce. Většina mobilních zařízení má nastavené vysoké rozměry viewport layoutu.

Tento problém řeší metatag viewport. Ten má 6 direktiv, v tomto projektu postačí pouze základní a nejdůležitější direktiva width. Ta umožňuje nastavit viewport na zadanou šířku, či na šířku zařízení. Asi nejlepší způsob, který byl také zvolen v tomto projektu, je nastavení viewportu na šířku zařízení, jak lze vidět na obrázku 4.3. Tento zápis způsobí, že se bude viewport layoutu obrazovky rovnat obrazovce zařízení.

A screenshot of a code editor showing a single line of HTML code: `<meta name="viewport" content="width=device-width" />`. The code is highlighted with a light blue background. To the left of the code, there is a small box containing the number 9.

Obr. 4.3 – Nastavení viewportu na šířku zařízení (Zdroj: autor)

Ostatní direktivy slouží například k nastavení výšky, zdali uživatel bude moci stránku přibližovat, jak moc může na stránce uživatel přibližovat či jak hodně může uživatel stránku zmenšovat.

4.4.2 Pořadí mediálních dotazů

V tomto projektu byl responzivní design implementován z desktopů směrem dolů. Primárním zařízením byl v projektu vzhled pro desktopová zařízení, tudíž byl nejprve vyvíjen stylový předpis pro tato zařízení. Následně byl web pomocí mediálních dotazů zjednodušován a přizpůsobován menším zařízením s menšími displeji.

Mediální dotazy byly vkládány přímo do hlavního stylového předpisu. Je to z důvodu, že u tohoto webu není zapotřebí velké množství kódu navíc, a tak soubor se styly nebude příliš velký. Navíc postačí pouze jeden http požadavek, což je pro mobilní telefony využívající mobilní síť výhodou.

4.4.3 Breakpointy

Breakpoint je rozměr displeje, který se používá pro vymezení určité oblasti, ve které bude web vypadat přibližně stejně. Většinou se breakpointy určovaly podle určitých standardních šířek. Tento způsob má bohužel svoje nevýhody. Hlavní nevýhodou je otáčení mobilních telefonů. Zařízení, které je otočeno na výšku může mít displej široký 320 pixelů, ale pokud je takové zařízení otočeno na šířku, šířka displeje se změní například na 480 pixelů. Následně záleží na obsahu webu, jestli se stránka rozsype nebo ne. Z toho důvodu se doporučuje vytvářet breakpointy podle obsahu webu a ne podle rozlišení běžných zařízení. Velkou výhodou u určování breakpointů podle obsahu je plovoucí layout kódovaný v procentech. V tomto případě lze snadno určit breakpointy pomocí prohlížeče. Pokud se responzivní design tvoří z desktopu směrem dolů, což je případ tohoto projektu, stačí pomalu zmenšovat okno prohlížeče. Když se obsah webu začne měnit a vypadat špatně, například když začnou objekty webu měnit svou pozici, text je špatně čitelný, vznikne příliš hodně prázdného místa a podobně, je to znamení pro vytvoření breakpointu.

V tomto projektu byl vytvořen první breakpoint, tedy ten nejúžší, na 418 pixelů. Původně bylo zamýšleno zvolit první breakpoint na šířku displeje 360 pixelů. Tato šířka je nejvyužívanější rozlišení v Evropě jak lze vidět na obrázku 4.1. Pod hranicí této šířky byl web takřka nečitelný. Na řádek se vlezlo pouze jedno slovo, z důvodu vystředění celého obalu webu a ostatním vnitřním a vnějším okrajům. Horizontální navigační menu se rozsypalo a zabíralo celé záhlaví stránky. Také obrázky, jimiž bylo doplněno záhlaví, byly v takto malém rozlišení nevhodné. Po odzkoušení funkčnosti designu a manipulaci s webem bylo rozhodnuto, že bude první breakpoint rozšířen na 418 pixelů. V této šířce se již odkazy horizontálního menu nevlezly vedle sebe a poslední z odkazu přeskakoval na nový řádek. Rovněž obrázky při této šířce neměly význam a zabíraly místo na úkor loga.

Z těchto důvodů byly provedeny úpravy, které se projeví při šířce displeje od 418 pixelů níže. Celému obalu byla nastavena šířka na 100%, následně také záhlaví a zápatí, jelikož je potřeba zaplnit celou šířku displeje. Doplnující obrázky ze záhlaví byly skryty a ponecháno bylo pouze logo kadeřnictví. Také výška záhlaví byla snížena, aby zabírala co nejméně místa na viditelné ploše displeje. Horizontální navigační menu by se na tak malou šířku nevlezlo a samotné odkazy v bloku by zabíraly hodně místa na viditelné části displeje. Proto bylo navigační menu s odkazy skryto a zobrazí se po kliknutí na tlačítko menu, které je k dispozici pouze při šířce 360 pixelů a méně. Následně byly provedeny úpravy hlavní části main, aby se obsah zobrazoval co nejlépe.

```

427 @media all and (max-width: 418px) {
428     #obal
429     {
430         width: 100%
431     }
432     #motiv, #vlasy
433     {
434         display: none;
435     }
436 }

```

Obr. 4.4 – Mediální dotaz a část stylů pro zařízení do 360 pixelů (Zdroj: autor)

Funkčnost tohoto designu byla následně otestována na webu <http://beta.screenqueri.es>, kde bylo odzkoušeno zobrazení na několika mobilních telefonech se šířkou displeje do 360 pixelů. Na všech zkoušených zařízeních byl design funkční a vše se zobrazovalo tak jak má. Rovněž byl design odzkoušen na fyzických zařízeních, kde byl také funkční a zobrazoval se korektně. Konkrétně šlo o displeje s úhlopříčkou 4,3, 5 a 5,5 palce. Na obrázku 4.5, který byl pořízen prostřednictvím webu www.deviceponsive.com, lze vidět zobrazení designu na displeji o rozlišení 320x480 pixelů. Vertikální posuvník je zde zobrazen pouze z důvodu, aby bylo možné odzkoušet design i v prostředí bez dotykových obrazovek.



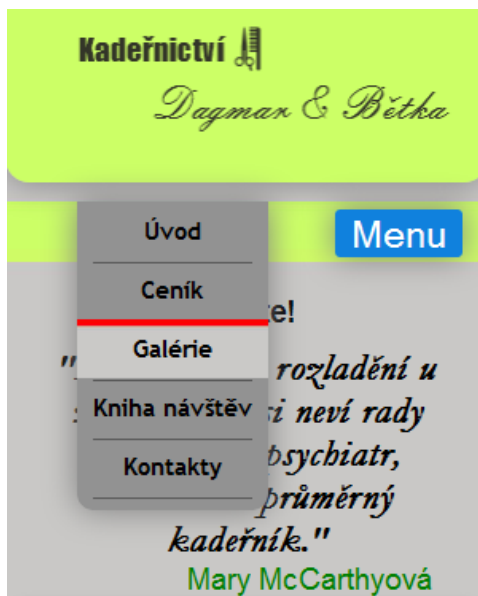
Obr. 4.5 – Test responzivního designu (Zdroj: www.deviceponsive.com)

Další breakpoint byl zvolen na šířku 768 pixelů. Jak vyplývá z grafu společnosti Stat Counter na obrázku 4.2, je tato šířka displeje nejpoužívanější v Evropě u tabletů. Tato šířka displeje je dostatečně široká pro zobrazení horizontálního navigačního menu. Při zarovnání menu doprava, tak jako tomu je u větších obrazovek, by nebylo dostatek místa pro všechny odkazy, a proto je menu vystředěno doprostřed stránky. Levý postranní panel s otevírací dobou by při rozlišení do této šířky zabíral spoustu místa, z toho důvodů byl umístěn pod hlavní část webu. Šířka celé stránky byla zvětšena na 95 procent, aby bylo více místa pro hlavní obsah. Po manipulaci s webem bylo zjištěno, že vzhled stránky v rozmezí šířky displeje od 418 do 768 pixelů není vyhovující. Šlo hlavně o odkazy v horizontálním navigačním menu a obrázky v hlavičce. Při nižší šířce displeje by odkazy začaly přeskakovat na nový řádek, což vypadá nevzhledně. Obrázky byly v tomto stavu rovněž nevzhledné a spíše na obtíž. Z těchto důvodů byl přidán další breakpoint, který byl určen na šířku 567 pixelů. Při této šířce se už spojení “knihy návštěv” nevedle sebe. Šířka displeje je ale stále dostatečně široká a horizontální navigační menu lze až na odkaz knihy návštěv dobře zobrazit. Proto byl upraven pouze tento odkaz a menu ponecháno na svém místě. Poslední breakpoint byl nastaven na šířku do 1200 pixelů. Vystředění hlavního obsahu, které bylo původně nastaveno na větší úhlopříčku monitoru, bylo po překonání breakpointu 567 pixelů nevyhovující a pro obsah bylo stále málo místa. Proto byla upravena šířka hlavního obalu na 90 procent až do šířky displeje 1200 pixelů. Na tuto šířku už nebylo potřeba vytvářet další breakpoint.

4.4.4 Navigace

Horizontální navigační menu lze použít u širších obrazovek, jenže u mobilních telefonů tato navigace není realizovatelná. Nabízí se způsob řešit navigaci pomocí vertikálního menu, kde jsou odkazy naskládány na sebe. Tento způsob řešení nevypadá špatně, ale u mobilních telefonů zabere spoustu místa a hlavní část webu, samotný obsah je poté skryt dole u spodní části displeje, či ještě níže. Tento způsob je nevyhovující, protože uživatel vždy vidí hlavně navigační menu a hlavní část, kvůli které se na webu pohybuje je skryta. Z těchto důvodů byla celá navigace skryta a pomocí javascriptu vytvořeno speciální tlačítko, které po kliknutí uživatelem zobrazí navigační menu. Navigační menu se po aktivování zobrazí vertikálně směrem dolů přes hlavní obsah, se stejnými styly, jako jsou použity u horizontálního menu pro širší obrazovky. Tlačítko je dostatečně velké a výrazné, aby se uživatelé na webu přehledně orientovali a nemuseli nic složitě hledat. Toto speciální tlačítko se samozřejmě zobrazí díky mediálnímu dotazu až od šířky displeje 360 pixelů a níže.

Díky tomu bylo ušetřeno více místa na viditelné ploše obrazovky mobilního telefonu a zbylo tak více plochy pro zobrazení hlavního obsahu webu.



Obr. 4.6 – Aktivované menu na mobilním zařízení (Zdroj: autor)

4.5 Responzivní obsah

U responzivního webu je důležité pracovat s obsahem stránky. Obsah, který vypadá dobře na monitoru počítače, nemusí být na mobilních zařízeních takřka čitelný. Proto je velmi důležité promyslet si, jaká část obsahu má přednost před jinou a v jakém pořadí se budou tyto části zobrazovat na odlišných zařízeních. To může být problém hlavně u velkých a hodně členitých webů. V tomhle projektu musel být řešen pouze postranní sloupec s otevírací dobou. Na velkých obrazovkách je tento sloupec výhodou, protože má uživatel otevírací dobu stále na očích, navíc vyplní plochu, která by jinak byla nevyužita. Na mobilních zařízeních ale tento sloupec překáží a zabírá místo hlavnímu obsahu. Proto byl tento sloupec přesunut až pod hlavní obsah a rozdělen na dvě malé tabulky umístěné vedle sebe. Podobné úpravy byly provedeny také v kontaktech, kde jsou umístěny kontaktní informace na kadeřnice vertikálně vedle sebe. Tyto kontaktní informace byly pro mobilní zařízení umístěny pod sebe.

Některý obsah na malých obrazovkách ztrácí význam a proto je vhodné takovýto obsah skrýt. Proto byly ze záhlaví webu odebrány dva doplňující obrázky, které by byly moc malé a zabíraly by místo názvu kadeřnictví.

4.6 Responzivní obrázky

Z hlediska výkonu se u responzivních webu musí řešit také obrázky. Některé obrázky mají smysl pouze u velkých monitorů, ale u mobilních telefonů jsou nežádoucí. V tomto projektu jsou takové obrázky pouze dva v záhlaví webu a jejich skrytí je vyřešeno prostřednictvím vlastnosti display a hodnoty none. Použitím tohoto způsobu se obrázky skryjí, ale prohlížeč je bude pořád požadovat a stahovat. Toto může být u velkých webu s hodně obrázky problém a v takovém případě je nutno využít javascript a určit od jaké šířky se tyto obrázky nemají načítat.

Obrázky jsou ve formátu svg (Scalable Vector Graphic), který je vhodný pro obrázky, jejichž rozměry se mění na základě velikosti obrazovky. Jejich výhodou je, že mohou dobře měnit velikost a přitom se nijak nemění velikost samotného souboru. Jedinou nevýhodou tohoto formátu je, že má nedostatečnou podporu v některých prohlížečích.

Na webu jsou dále umístěny fotografie kadeřnictví ve formátu jpg, u kterých je požadováno, aby byly zobrazeny na všech zařízeních. Tyto fotografie jsou pouze upraveny tak, aby měly co nejmenší velikost souboru.

5. Nasazení webové prezentace

5.1 Validátory

Aby byly stránky, co nejvíce optimalizované pro vyhledávače je velmi důležitá validita kódu. Pokud je stránka validní, roboti vyhledávačů dokážou stránku správně zpracovat. Díky tomu se zvyšuje viditelnost webové prezentace. Stránka je validní, pokud odpovídá platným standardům. V případě tohoto webu jde o specifikaci HTML5. Webové stránky musí být kódovány tak, aby neobsahovaly žádné chyby. K ověření validity stránek slouží takzvané validátory, které porovnávají kód stránek s příslušným standardem. V tomto projektu byl použit validátor přímo od organizace w3c.

This document was successfully checked as HTML5!		
Result:	Passed, 4 warning(s)	
Address :	<input type="text" value="http://www.kadernictvidab.wz.cz/"/>	
Encoding :	utf-8	<input type="text" value="(detect automatically)"/>
Doctype :	HTML5	<input type="text" value="(detect automatically)"/>
Root Element:	html	

Obr. 5.1 – HTML validátor (Zdroj: <https://validator.w3.org/>)

Stejně jako u HTML existují validátory také pro CSS kód. V případě tohoto webu byly stránky testovány podle standardu CSS3. Pokud jsou stránky validní, mají také vyšší pravděpodobnost správného zobrazení v prohlížečích. Na obrázku 5.2 lze vidět výsledek validace CSS kódu tohoto webu.

Validátor výsledků W3C CSS <http://www.kadernictvidab.wz.cz> (CSS level 3)

Blahopřejeme! Chyby nenalezeny!

Tento dokument ověřuje jako [CSS level 3](#) !

Obr. 5.2 – CSS validátor (Zdroj: <http://jigsaw.w3.org/css-validator/>)

5.2 SEO

SEO je zkratka z anglického názvu Search Engine Optimization, což by se dalo do češtiny přeložit jako optimalizace pro vyhledávače. Je to způsob tvorby webových stránek takovým způsobem, aby jejich obsah a forma byla vhodná pro internetové vyhledávače. Cílem je získat ve výsledcích vyhledávání co nejlepší pozici a tím zvýšit návštěvnost webu. Existují weby, které provádějí analýzu zdrojového kódu zaměřenou na SEO. Tyto analyzátoři se zaměřují na spoustu věcí, například na:

- hlavička dokumentu – správná deklarace, titule, popis stránky, použití klíčových slov,
- zdrojový kód – validita kódu, velikost kódu,
- sémantika a přístupnost – web neobsahuje vnořené tabulky a podobně,
- obsahová část – pouze jeden h1 nadpis, správné pořadí nadpisu, počet slov.

Úvodní strana » Zdrojový kód » Výsledek testu


Analýza zdrojového kódu

Adresa: www.kadernictvidab.wz.cz

- Datum testování: 28. 06. 2015
- Celkové hodnocení: 94 %

Popisné informace

Titulek	Kadernictví D&B
Popis	Kadernictví Šenov
Klíčová slova	kadernictví, holičství, účesy, Šenov, Šenov u Ostravy
Meta robots	all
Autor	Jakub Kohoutek
robots.txt	Existuje
Sitemap	www.kadernictvidab.wz.cz/sitemap.xml



Obr. 5.3 – Analýza zdrojového kódu (Zdroj: www.seo-servis.cz)

Při optimalizaci pro vyhledávače je rovněž důležitý soubor robots.txt, který musí být umístěny v kořenovém adresáři webu. Jeho název je povinné psát malými písmeny. Pokud soubor robots.txt existuje, roboti se nejdříve podívají do něj a až poté začínají prohledávat web. Roboti v souboru naleznou informace o tom, které stránky webu nemohou navštěvovat.

Zápis příkazů do souboru robots.txt je velmi jednoduchý. Roboti jsou nazýváni User-Agent a jejich omezení se přidává pomocí příkazu Disallow. Roboty je možno vypisovat postupně například Google robot má název Googlebot, seznam robot zase SeznamBot, nebo pomocí hvězdičky která nahrazuje všechny roboty. Za příkaz Disallow se píše název adresáře, do kterého mají roboti zákaz.

V tomto projektu není potřeba robotům zakazovat navštívení, či stahování jakékoli stránky na webu, jelikož neobsahuje žádné citlivé informace, jako například placené články, nebo stránky se správou webu. Obsah souboru robots.txt proto vypadá následovně:

User-Agent: *

Disallow:

Sitemap: <http://kadernictvidab.wz.cz/sitemap.xml>

Tento zápis znamená, že všichni roboti mohou prohledávat obsah celého webu. Prázdný nebo neexistující soubor robots.txt má stejný význam. Je ale doporučováno soubor robots.txt vždy vytvořit, aby nedošlo k chybě s indexací v situaci, kdy není definovaná chybová stránka 404.

Navíc je v souboru uvedena adresa na další důležitý soubor sitemap.xml, aby jej roboti našli a mohli zpracovat. Soubor sitemap.xml je mapa webu, která napomáhá robotům v prohledávání webu. V tomto projektu byl použit sitemap generátor dostupný na stránkách www.xml-sitemaps.com.

Indexování obsahu a sledování odkazů je v tomto projektu robotům rovněž povoleno, a to pomocí meta tagu robots. Ten se zapisuje do hlavičky webu do značky meta s přístupovou hodnotou all. Zápis vypadá následovně:

<meta name='robots' content='all'>

5.3 Jak dostat web do vyhledávačů

Pro vyhledávače stránky shromažďují takzvaní roboti. Ti se pohybují Internetem pomocí hypertextových odkazů, což znamená, že pokud na danou stránku neexistuje odkaz, robot jí nenajde a stránka se nedostane do vyhledávačů. Z tohoto důvodu je potřeba vytvořit zpětné odkazy na web, aby je roboti mohli najít. První způsob jak vytvořit zpětný odkaz je umístit

odkaz daného webu na jiný web, například prostřednictvím výměnné reklamy. Druhá možnost jsou katalogy vyhledávačů. Tato možnost byla využita pro vytvoření zpětných odkazů v tomto projektu. Každý vyhledávač má svůj internetový katalog, kde lze web zaregistrovat do příslušné kategorie. Tento web byl zaregistrován do katalogu českého vyhledávače seznam a do katalogu vyhledávače google. Přidání webu do katalogu je velmi jednoduché, adresa webu se vloží do přidávacího formuláře a po několika dnech by měl robot adresu zaregistrovat. Stejným způsobem lze adresu přidat také do googlu. Dále platí, že čím více bude web optimalizovaný pro vyhledávače, tím výš se bude umisťovat ve výsledcích vyhledávání.

✓ Přidání URL <http://kadernictvidab.wz.cz> proběhlo v pořádku

Adresa webu

Obr. 5.4 – Přidání adresy webu do katalogu seznamu (Zdroj: autor)

5.4 Webhosting

Webhosting poskytuje zázemí pro webové stránky. Poskytuje také mnoho služeb pro obsluhu stránek. Existuje spousta druhů webhostingu od těch nabízených zdarma až po placené. K tomu aby mohl být web umístěn na internet je kromě webhostingu nutná doména. Doménu je možno zakoupit přímo v balíčku s webhostingem.

Pro tento projekt byl vybrán webhosting webzdarma.cz nabízený, jak již z názvu vyplývá, zdarma. Pro prvotní spuštění a otestování webu byl shledán tento způsob jako nejvhodnější. Webzdarma nabízí adresu ve tvaru domény 3. řádu s možností výběru z 18 domén 2. řádu. Pro tento web byla zvolena adresa kadernictvidab.wz.cz. Adresa obsahuje klíčové slovo kadernictví, následují iniciály názvu kadernictví, aby byla adresa kratší. Rovněž doména druhého řádu, byla zvolena nejkratší, tedy wz. Ve výběru byly domény 2. řádu například borec, kvalitne, prodejce, webzdarma. Při použití některé z těchto domén by adresa byla už příliš dlouhá.

Webzdarma ve své bezplatné verzi dále nabízí například:

- uživatelský prostor 200-500 MB,
- databázi MySQL 5.5 o velikosti 20 MB;
- podporu skriptu PHP 5.5
- zákaznickou podporu.

Daní za bezplatné poskytování webhostingu je u webzdarma nutnost umístit na svůj web reklamu. Ta se do webu vkládá automaticky sama při načítání stránky. Na webu webzdarma si lze zvolit kde bude reklama umístěna. Jsou tři varianty, první je umístění reklamy nahoru na úvodní stránku, na ostatních se poté nezobrazuje, další varianta je umístění reklamy dolů na všechny stránky webu a poslední je umístit reklamu do specifického souboru a poté ji vhodně umístit na stránku. Poslední varianta je pro web asi nejvhodnější, ale také nejkomplikovanější. V případě zájmu webzdarma nabízí i placenou verzi webhostingu, která nabízí více funkcí.

6. Závěr

Tato práce byla rozdělena do šesti hlavních kapitol počínaje úvodem, kde je popsán cíl práce a konče závěrem. V druhé kapitole je popsána teorie a metody tvorby webových stránek. Do této kapitoly byly zahrnuty nejdůležitější části tvorby webových stránek. Od funkce internetu přes používané technologie s popisem jejich nejnovějších verzí až po software určený k samotné realizaci webu. Ve třetí kapitole je řešena koncepce webové prezentace od samotného plánování až po kód stránek. Byl vybrán nejvhodnější software pro tvorbu daného webu a vytvořen produkční plán k naplňování projektu. Následně bylo popsáno rozvržení hlavních částí na stránce, celkový design webu a nakonec samotné kódování webu a vzhledu. Čtvrtá kapitola je zaměřena na responzivní design a jeho implementaci v projektu. Bylo popsáno proč je responzivní design důležitý a jak může uspokojit potřeby uživatele. Následně bylo provedeno naplňování použití responzivního designu vzhledem k danému webu, provedení analýz a výběr testovacího prostředí pro odzkoušení funkčnosti celého designu. Poté bylo přistoupeno k samotné implementaci, byly vybrány co nejvíce responzivní jednotky a následně web upravován pomocí mediálních dotazů tedy stěžejní části responzivního designu. K posledním částem této kapitoly patří popis rozložení obsahu na stránce při použití responzivního designu a problematika obrázku u responzivních webů. Pátá kapitola byla věnována samotnému nasazení webové prezentace. Byly provedeny testy validity kódu a také analýza zdrojového kódu k optimalizaci pro vyhledávače. Následně byla webová prezentace zaregistrována do katalogů dvou nejznámějších českých vyhledávačů. Na samotný závěr byl vybrán webhosting nabízený zdarma, který je dostatečný pro funkčnost webu a vhodný pro jeho odzkoušení v provozu.

Hlavním cílem této práce bylo vytvořit webovou prezentaci pro kadeřnictví s využitím responzivního designu. To se také povedlo, kadeřnictví má novou webovou prezentaci, která jasně plní svůj prezentační a informační účel. Web je přehledný a jasný, neobsahuje žádné nadbytečné prvky, které by uživatele rušili či dezorientovali. Responzivní design byl využit, a proto je webová prezentace přívětivá k většímu počtu zařízení. Kód webu byl psán podle nejnovějších standardů a prošel validací, celý web je také optimalizovaný pro vyhledávače. Webová prezentace byla zaregistrovaná do katalogů vyhledávačů a již po pár dnech ji lze vyhledat. Do budoucna by se dal očekávat růst pozice webu ve výsledcích vyhledávání. Cíl bakalářské práce byl tedy naplněn.

Seznam použité literatury

Tištěné zdroje

DOVER, Danny a Erik DAFFORN. SEO: optimalizace pro vyhledávače profesionálně. Vyd. 1. Brno: Zoner Press, 2012, 400 s. Encyklopedie webdesignera. ISBN 978-80-7413-172-1.

GASSTON, Peter. *Moderní web*. 1. vyd. Překlad Ondřej Baše. Brno: Computer Press, 2015, 240 s. ISBN 9788025143452.

HAUSER, Marianne, Tobias HAUSER a Christian WENZ. *HTML a CSS: velká kniha řešení*. Vyd. 1. Brno: Computer Press, 2006, 912 s. ISBN 80-251-1117-2.

KADLEC, Tim. Responzivní design profesionálně. Vyd. 1. Brno: Zoner Press, 2014, 246 s. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.

PÍSEK, Slavoj. *HTML: začínáme programovat*. 4., aktualiz. vyd. Praha: Grada, 2014, 181 s. Průvodce (Grada). ISBN 978-80-247-5059-0.

PONKRÁC, Miloslav. PHP a MySQL: bez předchozích znalostí. Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1758-3.

PROCHÁZKA, David. PHP 6: začínáme programovat. 1. vyd. Praha: Grada, 2012, 183 s. Průvodce (Grada). ISBN 978-80-247-3899-4.

PROKOP, Marek. *CSS pro webdesignery*. Vyd. 2. Brno: CP Books, 2005, 288 s. ISBN 80-251-0487-7.

WEMPEN, Faithe. *HTML a CSS: krok za krokem*. Vyd. 1. Brno: Computer Press, 2007, 324 s. Krok za krokem (Computer Press). ISBN 978-80-251-1505-3.

Elektronické zdroje

Dušan Janovský. Syntaxe XHTML [online]. Jak psát web [22. 4. 2015]. Dostupné z: <http://www.jakpsatweb.cz/html/xhtmll.html>

Jiří Kosek. Historie a vývoj HTML [online]. HTML guru [22. 4. 2015]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>

Josef Žáček. CSS pseudo elementy [online]. Josef Zacek Web Development [24. 4. 2015]. Dostupné z: <http://www.josefzacek.cz/css-pseudo-elementy/>

Manuel Lemos. PHP 7 Features and Release Date - PHP Classes blog [online]. PHP Classes [20. 5. 2015]. Dostupné z: <http://www.phpclasses.org/blog/post/242-PHP-7-Features-and-Release-Date.html>

Pavel Kasík. Proč Google Chrome předběhl IE a stal se nejpoužívanějším prohlížečem [online]. Technet [22. 5. 2015]. Dostupné z: http://technet.idnes.cz/google-chrome-predbeh-ie-internet-explorerer-fzt-/sw_internet.aspx?c=A120521_150519_sw_internet_pka

World Wide Web Consortium. W3C: W3C mission [online]. W3C [20. 4. 2015]. Dostupné z: <http://www.w3.org/Consortium/mission>

Seznam zkratek

ARPANET - Advanced Research Project Agency Net

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

JPG - Joint Photographic Experts Group

PHP – Hypertext Preprocessor

SEO – Search Engine Optimization

SQL - Structured Query Language

SVG - Scalable Vector Graphic

TCP/IP – Transmission Control Protocol/Internet Protocol

URL – Uniform Resource Locator

W3C – World Wide Web Consortium

XHTML – Extensible HyperText Markup Language

XML - Extensible Markup Language

WHATWG - What You See Is What You Get

WWW – World Wide Web

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 10. 7. 2015



Jakub Kohoutek

Seznam příloh

- Příloha č. 1 – Ukázka galerie
- Příloha č. 2 – Ukázka návštěvní knihy

Přílohy

Příloha č. 1

